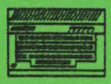
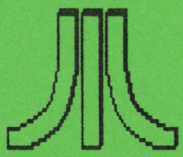
T — Y — N — E — & — W — E — A — R



TYNE & WEAR



ATARI [8]-BIT USER GROUP

Newsletter of TWAUG

Software
Editorial
Buy & Sell
Hardware

TYNE&WEAR
ATARI
USER GROUP

Reviews
Help line
Section
Repair Info

Public Domain Library

ISSUE #16

JULY/AUGUST 1995

U — S — E — R — G — R — O — U — P

TWAUG NEWSLETTER

BRING YOUR EIGHT UP TO DATE
with power products from
COMPUTER SOFTWARE SERVICES

THE BLACK BOX

The BLACK BOX is an add-on board for the Atari 600XL,800XL and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built into the BLACK BOX so no adapter boards are necessary. A cartridge port is available on the board itself for 130XE users.

The BLACK BOX provides many unique and useful functions. The four primary functions are:-

* RS-232 serial modem port
* Parallel printer port
* SASI/SCSI hard disk port
* Operating System enhancements

The BLACK BOX is $199.95 for the basic unit, and $249.95 with an onboard 64K printer buffer.
Shipping and Handling extra.

THE BLACK BOX ENHANCER

A must for all BLACK BOX owners. The BLACK BOX ENHANCER is a plug-in module for your BLACK BOX, enhancing the printer functions and adding an instantly available, full featured sector editor!

Installation of the BLACK BOX ENHANCER requires one simple solder connection. Only $49.95 plus shipping/handling.

THE FLOPPY BOARD

Our latest and greatest product. The FLOPPY BOARD is an add-on expansion board for the BLACK BOX interface. It allows the use of the same inexpensive floppy drive mechanisms used in IBM computers. The FLOPPY BOARD is the first floppy drive interface to support "high density" floppy drive mechanisms in either 5.25 inch or 3.5 inch. Built into the FLOPPY BOARD are our BLACK BOX ENHANCER and a version of our SUPER ARCHIVER to allow copying of protected disks for 3.5 inch format. Included with the FLOPPY BOARD is our program to read and write to IBM or ST formatted disks. This makes the FLOPPY BOARD the best way to transfer files to and from your 8-bit.

The FLOPPY BOARD is only $149.95 plus shipping & handling.

THE MULTIPLEXER

This device brings the power and flexibility of larger systems to your 8-bit. The Multiplexer is a collection of cartridge interface boards that allow up to 8 Ataris to read and write to the same drives (typically a hard disk), access the same printer(s), and talk to each other. It is the first practical networking system for the Atari 8-bit computer.
One "master" computer (any 8-bit) is equipped with the master Multiplexer interface.
Then up to 8 "slave" computers hook up to this master, each having their own slave interface. The "common" peripherals (things that are to be shared) are connected to the master. On each slave, all disk and printer I/O is routed through the master, so no extra disk drives are needed.

The Multiplexer sells for $199.95 for a master and two slave units with cable. Additional slave units are $89.95 each, plus shipping/handling.

THE SUPER ARCHIVER II

The SUPER ARCHIVER II edits and copies all enhanced density programs plus retains all the features of the SUPER ARCHIVER.

The SUPER ARCHIVER II is only $99.95 plus shipping $ handling. NOTICE: if you already have THE SUPER ARCHIVER you may upgrade to S.A.II for only $29.95 plus shipping/handling. Software only.

THE BIT WRITER

The Super Archiver BIT WRITER is capable of duplicating even the "uncopyable" Electronic Arts and Synapse Syn-series, which employ 34 full sector tracks. The BIT WRITER must be used with the SUPER ARCHIVER

The BIT WRITER is only $79.95 plus shipping/handling.

THE ULTRA SPEED PLUS OS

The Operating System that should be in every XL/XE computer! The Ultra Speed Plus puts unbelievable speed and convenience at your fingertips.

Use any DOS to place Ultra Speed formats on your disks (with XF551 or modified 1050 drives), reading and writing at this speed with most programs. This high speed mode can be turned off for maximum compatibility.

Four simple solder connections are required for installation if your machine has a socketed OS ROM. The Ultra Speed OS is only $69.95 plus shipping/handling.

For more information on these and other 8-bit products:

CONTACT
COMPUTER SOFTWARE SERVICES
PO BOX 17668
ROCHESTER, NEW YORK 14617
USA

ORDERING LINE: (716) 429-5639
FAX: (716) 247-7158
BBS: (716) 247-7157

EDITORIAL

Who to blame!!!

John Matthewson
David Ewens
Max Gerum

TWAUG Newsletter has been produced entirely on an 8-bit machine. The lay-out of the pages is done using TextPro 4.54 and printed using Daisy-Dot 3 print processor. I am using the 65XE machine upgraded to 1-Meg attached is the Black Box with a 40-Meg Hard Drive, also three PBI drives, one 3 1/2 Hi-Low density drive, one Low 5 1/4 and one 5 1/4 Hi-density drive and one 1050 drive enhanced with US Doubler and Happy Board. The master copy is printed on a Citizen 120D+. The newsletter is reduced to A5 with a photocopier and all the issues are produced using the photocopier.

The contribution fees for home and abroad:

HOME	1 COPY	£2.00
-DO-	6 COPIES	£11.00
EUROPE	1 COPY	£2.20
--DO--	6 COPIES	£12.50
ELSEWHERE	1 COPY	£2.50
---DO----	6 COPIES	£14.00

REMINDER:

The book--The Complete and Essential Map--anyone unable to pay the full amount of the cash price can pay in instalments, whenever and whatever you can afford, at no extra charge.

The next issue will be ready by mid-September.

CONTENTS

EDITORIAL 3

MODEM PROBLEMS
from Z*MAG 4

ERROR 164
unknown 5

JOYSTICK PORT
by Chuck Grimsby 6

SPARTADOS X
by Doug Wokoun 8

REPAIRING FUNCTION KEYS
by Dean Lowery 10

LACE ADVERT 11

GAMES REVIEWS
by Kevin Cooke 12

PRC CONSTRUCTION SET
by Bill Walraven 15

APOLOGY from MAX 16

SEARCHING FOR BOBBY FISHER
Chess Games on Atari by Ed Hall 17

256 MEMORY UPGRADES
by Bob Woolley 18

TWAUG ANNOUNCEMENT 22

CRACKING THE CODE
by Keith Mayhew 24

ALTERNATE REALITY 34

TWAUG ANNOUNCEMENT 38

OL'HACKERS ADVERT 39

LETTER SECTION 40
DISK CONTENT 41
FOR SALE SECTION 42

MICRO DISCOUNT 43

ATARI CLASSIC
PROGRAMMERS CLUB
and CHAOS! COMPUTERS 44

MODEM PROBLEMS

This article has been taken from Z*MAG issue 205. It deals with problems encountered with modems, and discusses some myths about modems.

LINE NOISE

Many people have left messages on my bulletin board asking me why there are so many 'garbage' characters on their screens and why file transfers are riddled with errors. These garbage characters are really line noise and can be introduced in many different places. Pure noise is a decimal 255 (FF in hex), but most line noise is not 'pure'. It usually comes in as something less than 255, like maybe a 251 (a character that looks like this "). Ever see that one before? Yup, so have I!

One of the more common and familiar introduction points of line noise is in the telephone company's system and even here there are several ways noise is introduced. A signal is routed through multiple stations before it eventually makes it to the other end and some of these stations aren't exactly new. Older areas may have older, less sophisticated equipment that is more apt to be affected by ambient noise. This is one reason some people continue to have noise problems even after hanging up and calling back multiple times. Also, a given physical connection at one of these junctions may not be up to snuff. If your particular bout of line noise is solved by hanging up and calling back, then it's probable that you were previously connected through an intermittent or 'dirty' connection. Some of these trunk lines (large, multi-area that has a lot of ambient RFI (Radio frequency Interference) present although this is not usually the case.

It is possible that the problem is being caused at this end, but not if the problem goes away when you call back and the line is clean -or- if you are one of a very few users experiencing noise problems. You may say that you are not having problems with other boards.....in which case the problem is more than likely the route that your call takes to get here. You may be going over micro-wave or through buried cable which for some reason are sub-standard. No matter how many times you call, you will probably be routed over the same path. Microwave problems are sometimes the hardest to track down because they can cause intermittent problems. Some interference only occurs during certain times of the day or week.

Another common noise introduction point is in your home. Most residential homes have televisions, radios, microwave ovens, VCR's, and if you are reading this, a micro-computer. All these devices radiate radio waves that can (and often do) get into the phone lines and cause noise. Electric motors and technical dimmer controls can introduce noise into the electrical wiring in your house and cause problems. If your line noise problem does not go away after repeated hanging up and calling back, then you may be suffering from one of these household problems. If you are suffering from this problem, you can take steps to eliminate it. First of all, turn off EVERYTHING except the fridge (If it IS the fridge, then you're SOL. Can't live life with your ice box unplugged) and see if the noise persists. If it goes away, then start turning things back on, checking the computer each time until you see the noise start up again. It may be that a single device is not bugging you but several devices plotting together to annoy you. This elimination tournament may take a while.

Another area to check is your wiring at the computer. Use noise suppressors on your power connections to both the PC and the modem (if external). Use a shielded RS-232 cable to connect your modem to the PC. Ribbon cables (especially long runs of it) are great antennas and will cause problems. Re-route the RS-232 cable so it does not run next to the PC power supply or any other transformer.

And now a little discussion about the modem itself. First of all, I'd like to clarify a commonly misused

MODEM PROBLEMS continued

term - BAUD. The term "Baud" is actually a man's name - J.M.E. Baudot (Pronounced: Baw-doe) a French Telegraphy expert. 1,200 and 2,400 Baud is NOT the same as 1,200 and 2,400 BPS (Bits Per Second). The usage of "Baud" to describe line speed in terms of data through-put is incorrect. 1,200 and 2,400 BPS modems both operate at 600 Baud. Basically, without getting to technical, a Baud is a "blip" of information. 1,200 BPS modems use four states per blip (or Baud) and 2,400 BPS modems use sixteen states per blip. If you want more information on what Baud and BPS mean and a full explanation of how data is actually represented and transferred by the modem, please refer to PC Magazine Volume 6, Number 9 (May 12, 1987).

Modems operating at 2,400 BPS are much more intolerant of line noise than are modems operating at 1,200 BPS. Conversely, modems capable of 2,400BPS operate better at 1,200 BPS than do 1,200 BPS only modems. If you are being hopelessly attacked by noise at 2,400 BPS, try calling back at 1,200 BPS. It's very possible that the noise will be greatly reduced or disappear altogether. I know, you didn't buy a 2,400 BPS modem just to retard it to 1,200 BPS. The brand of the modem plays a part in the immunity to line noise. Some modems can digest more noise (lower signal-to-noise radio) than others. PC Magazine (same issue mentioned above) ran a test on 87 different modems. You might check the results to see how your modem ranks. Most 2,400 BPS modems operating at 1,200 BPS have approximately -8 to -10 db error threshold while the same modem has about -16 to -20 db threshold operating at 2,400 BPS. For this reason, line quality is much more critical at 2,400 BPS operation.

Additionally, a friend of mine who runs a bulletin board from their office has been plagued with line noise problems at 2,400 BPS but very little noise at 1,200 BPS. The culprit is the office's centralized telephone system. Many office buildings have a given number of trunks that actually enter the building while there may be many, many more extension within the building. These types of telephone systems have their own controllers and line assignment devices and are frequently not as high in quality as a hard-wired MaBell (or GTE) line. The acceptable signal-to-noise ratio in some of these inter-office phone controllers are lower than necessary for reliable 2,400 BPS operation but not too low for 1,200 BPS.

If you get transmission errors while downloading or uploading a file, don't fret. The Xmodem (or whatever protocol) incorporates an error checking/correction mechanism that automatically detects and corrects any errors that may occur during transmission. The very fact that Xmodem reported the error in the first place means that he caught it and corrected it. The only errors you have to worry about are the ones that Xmodem does NOT report. Any reported error has already be corrected. Xmodem, especially the CRC flavoured one, is a very reliable file transfer protocol. Even if you got 100 errors during transmission, chances are still pretty slim that the file got corrupted. Occasionally, a file will be corrupted after transfer, but many times this may be due to a bad ARCing of the file or perhaps a disk error that may have occurred sometime during the files' past.

ERROR 164.

There is NO program that can GUARANTEE to fix a disk when you get an error 164.

An error 164 usually results when you have two files trying to use the same spot on the disk. This usually happens because you have saved one file to disk, done something illegal, and then saved a second file to the same disk. The second file is probably okay and is probably completely accessible. But the first file is simply GONE because the second file has written over the top of it. Period.

Can you recover part of the dama-

ERROR 164 continued

ged file? Possibly. But I would suggest that unless it is a text file (e.g., a word processing data file or possibly a LISTed -- NOT SAVEd -- program) the effort is bound to fail. Most SAVEd files, whether binary files or BASIC programs, simply CAN NOT be restored if there are missing pieces. Sorry.

Having said all that, what CAN you do with a damaged disk? Well, the DISKFIX utility that is part of DOS 2.5 will at least TRY to recover as much of a disk as it can. But if it decides a file is damaged beyond repair, it simply removes the file from the directory! So I would recommend making a sector copy of any damaged disk before attempting to use DOS 2.5's DISKFIX.COM program.

You can get DISKFIX here on CIS, in the DL's (DL 3, file DISKFI.*). But I personally recommend that you send off to Atari for a copy of not only a disk with DOS 2.5 but also a really good manual. The manual alone is worth the $10 or so that Atari charges.

Finally: The Atari DOS manual suggests this program to recover as much as possible of a damaged file, so long as the file is not bigger than available RAM in your machine:

```
10 PRINT "INSERT DAMAGED DISK
THEN GIVE NAME OF FILE TO
RECOVER ";
20 F=FRE(0)-300 : DIM BUF$(F),
FILE$(20)
30 INPUT FILE$
40 OPEN #1, 4, 0, FILE$
50 TRAP 100
60 FOR I=1 TO F : GET #1, B :
BUF$(I)=CHR$(B)
70 NEXT I
80 PRINT "FILE TOO BIG"
90 END
100 REM GET HERE ON ERROR...
110 TRAP 120 : CLOSE #1
120 PRINT "INSERT FORMATTED DISK
THEN GIVE NAME TO SAVE FILE TO ";
130 INPUT FILE$
140 OPEN #2, 8, 0, FILE$
150 PRINT #2; BUF$ ;
```

You could be neat and add line 160:

```
160 END
```

Finally, line 150 might be safer done as:

```
150 FOR J=1 TO I-1 : PUT #2,
ASC(BUF$(J)) : NEXT J
```

OOPS...One more finally.

In one of my articles in COMPUTE, I discussed probable causes of messed up disks. I noted that I personally have virtually NEVER had a disk messed up by DOS. Reason: I never never never change disks unless the program tells me to do so. If I am changing disks while using BASIC, I always type "END" before doing so. In desperation, I will hit RESET before changing.

The single most common cause of disk crashes is inserting a new disk while a file (or files) is still OPEN for output on the first one. This can happen with word processing programs, data bases, etc., etc. ALWAYS WAIT for the program to tell you it is time to swap disks. ALWAYS use the menus to get to the "disk change" point. NEVER just yank a disk and plunk in a new one.

JOYSTICK PORT

By Chuck Grimsby

An article on getting the most out of the joystick ports

As every Atari Basic programmer knows, the joystick port can be used to produce nine different actions or commands (excluding the center or 'null' position), utilizing the STICK(x) and STRIG(x) commands. The numbers your programs look for are:

value	stick position
14	UP
13	DOWN
7	RIGHT
11	LEFT
6	UP RIGHT
5	DOWN RIGHT
9	DOWN LEFT
10	UP LEFT
15	CENTER (NULL)
0	FIRE,(USING STRIG(0)
1	NOT FIRE

You may have noticed that there are

JOYSTICK PORT continued

some numbers missing from that list, and from all lists that show you how to use the STICK(x) command. Where are the numbers 0-4, 8 and 12? Well, actually those numbers are there and are readable, but you can't use a normal joystick to produce them. You either need a numeric keypad (like the old Atari CX-85) or a special 'joystick' consisting of buttons in place of a single stick. I built myself a special joystick to use as a non-moving mouse (my desk space is VERY limited) and discovered I had also created a joystick that would produce those non-standard numbers. My brother has dubed this device a 'Dead Mouse' and it has proved to be very handy. It also works great as a very accurate joystick for MicroPainter. The new STICK(x) list using the Dead Mouse looks like this:

VALUE	BUTTON(S) PRESSED
0	UP DOWN LEFT RIGHT
1	DOWN LEFT RIGHT
2	UP LEFT RIGHT
3	RIGHT LEFT
4	UP DOWN RIGHT
5	DOWN RIGHT
6	UP RIGHT
7	RIGHT
8	UP DOWN LEFT
9	DOWN LEFT
10	UP LEFT
11	LEFT
12	UP DOWN
13	DOWN
14	UP
15	NONE (NULL)

The numbers produced through the Dead Mouse can also be used to simulate the numeric keypad IF you have the proper AUTORUN.SYS file AND press the FIRE button with the other keys.

The following list shows the functions that the Dead Mouse key presses will return. Remember to ALWAYS press the FIRE button as well.

FUNCTION	DEAD MOUSE KEYS
DELETE	UP DOWN LEFT RIGHT
YES	UP DOWN LEFT
NO	UP DOWN RIGHT
-	NONE (NO KEYS PRESSED)
+ENTER	UP
0	UP DOWN
1	DOWN LEFT
2	UP LEFT
3	LEFT
4	DOWN LEFT RIGHT
5	UP LEFT RIGHT
6	RIGHT LEFT
7	DOWN RIGHT
8	UP RIGHT
9	RIGHT

The construction of the Dead Mouse is simple, due to the fact that every joystick made actually uses four buttons on the inside activated by moving the stick. All you need to make the Dead Mouse are five momentary contact buttons, a female D9 connector, a six conductor cable, and a project box. You can save yourself some time and trouble by using the cable from an old broken joystick, or buying a joystick extension cord and cutting off one end instead of making a new cable.

Start by drilling five holes for the buttons in the lid of the project box a little bit larger then the size of the buttons, and one hole in the side of the box for the cable to the computer.

Next mount and secure the five buttons in the holes and pass the cable through the hole in the side of the box. Make a knot in the cable on the inside so the cable wont pull out.

Now solder the wire from the cable to the switches following the table below. NOTE: Solder to ONLY ONE SIDE OF THE BUTTON!!

```
u 1 2 3 4 5 u
 O O O O O /
u O O O O /u
  6 7 8 9
```

PIN #	BUTTON
2	RIGHT
3	LEFT
4	DOWN
5	UP
9	FIRE

Now solder pin #7 to the other side

JOYSTICK PORT cont.

of ALL the buttons. This is the common or "ground" line. Put the lid on your box, and your Dead Mouse is ready to use. From experience, the Dead Mouse is a very poor joystick, Don't even bother to try and use it for game playing. It is ,however, a more professional looking device for use as a mouse than a joystick, and a accurate drawing tool for MicroPainter.

*** This is a very easy project, but as usual be careful and make sure of what you're doing. I can't be held responsible if anything goes wrong. Ed. ***

THE SPARTADOS X

review by Doug Wokoun

The SpartaDOS X cartridge is the latest incantation of SpartaDOS for the 8-bit Atari and very possibly the most powerful Disk Operating System available for any 8-bit computer.

The SpartaDOS X cartridge consists of 64K of ROM, with 48K (or 6 cartridge banks) formatted into a ROM-disk, and the remaining 16K used as the main DOS core. The ROM-disk contains files and drivers used by the system and SpartaDOS X versions of several utilities found in the SpartaDOS ToolKit. It also contains a very versatile ARC utility package.

Some of the new features of SpartaDOS X (referred to as SDX):

1. built in, memory resident FORMAT utility. Old versions of SpartaDOS could only initialize Atari format disks using 'AINIT'. To initialize a SpartaDOS disk required the loading of a program called 'XINIT'. Now, any time an XIO #254 call is made, the SDX format menu is brought up. With this, you can select a variety of disk densities and types. It will also allow "1-second" formatting by simply rewriting the root directory on a formatted disk.

2. High speed disk I/O with U.S. Doubler, Atari XF551, and Indus GT

THE SPARTADOS X cont.

disk drives.

3. New file loader supporting relocatable files (certain disk based commands can be held in memory and later removed) and symbol linking.

4. Probably the lowest MEMLO of any DOS. The DOS can load drivers under OS-RAM, into extended memory on an XE or at MEMLO on an 800.

5. Environment variables: user definable PROMPTs, search PATHs, parameter passing on batch files, and a CARtridge or BASIC memory save capability will retain programs even if the machine is shut off.

6. The ability to go from a cartridge to internal BASIC without rebooting. The CAR command enters the external cartridge, "BASIC" enters internal BASIC. You can go from Turbo BASIC XL to Atari BASIC to BASIC XE without rebooting! (with some provisions)

7. Support of up to 1 Meg internal memory as a RAMDisk.

8. "Persistent" batch files. Continued batch file processing even after loading binary programs.

9. Fast, powerful, versatile ARC utilities. Supports ALF files. With these, you can Add files to an ARChive, Move (delete after Adding), Freshen (update files by date), Update (Freshen with Add capability), Delete files from an ARChive, View files in ARC, eXtract files, and Print ARC'd files to screen. The ARC utilities also support password encryption and can function with the screen off to increase speed. Also, all files are sorted in alphabetical order when added to the ARChive.

10. A new MENU program very similar to the MS-DOS XTREE.EXE program. This program allows multi-file operations and displays the entire directory tree, so files anywhere on a disk can be accessed easily.

11. Command compatible with MS-DOS. Directory commands have several

THE SPARTADOS X continued

aliases. CWD from disk based SpartaDOS can also be accessed as CHDIR, or CD from SDX.

12. Drives can be referred to by letter or number.

13. Drives can be remapped. D1: can be SWAPped with D2:, etc. and from that point on, any referrences to D1: will be sent to D2: and vice versa.

SDX can be configured to take advantage of different hardware. A file placed on D1: called CONFIG.SYS is used for this, or the default configuration can be used. SDX can be configured to use OSRAM, or an extended bank of memory for its drivers. With the right setup, MEMLO can be pushed to below memory location $1000!

SDX uses a series of drivers to control most disk functions. SPARTA.SYS is the main driver and must be installed. 'DEVICE SPARTA' is used in the CONFIG.SYS file to do this. The number of sector buffers and file buffers can be controlled by passing parameters to this driver. Another driver is ATARIDOS.SYS used to read Atari DOS 2.x disks. Not installing this driver saves memory, but then Atari DOS disks cannot be read. The SDX cart also contains a RAMDisk driver which can be used to install up to 3 RAMDisks of any size. An INDUS.SYS driver is used to program the INDUS GT to operate at high speed. There are also two clock drivers, used depending on whether or not you have an R-Time 8 cartridge.

A major change with the X cart is the way devices are addressed. Since ICD wanted drives to be addressed by letter or number, conflicts would have occured with existing devices. Also, ICD wanted SDX to be more similar to MS-DOS, so those conventions were adopted. E: has become CON:, P: has become PRN:, and D1: D2: and D3: are A: B: and C:. Switching between an IBM machine and SpartaDOS X is much easier with these changes.

Another feature of SDX is its I/O redirection. With this, you can send the output of a program to another device. Ex: DIR >>PRN: would do a directory, but the results would be sent to the printer. Also, you can use a file to "feed" a program with input redirection. Ex: BASIC <<file.ext would call up internal BASIC and send it file.ext as if the contents of that file were being typed into the machine. This would be used in place of batch files because you can no longer send input to BASIC from a batch file.

SDX recognizes two new file attributes in addition to protected, hidden and archive. Hidden files do not appear in the directory, and archive is used to mark files for backup. This is normally used with a hard disk backup program. When a file is updated, the archive bit is cleared, telling a program like Flashback that the file needs to be backed up. All of these are set with the ATR or ATTRIB commands (same thing). You can also scan directories for files with certain attributes.

Two new commands, PEEK and POKE make many operations easier. Instead of going to BASIC to execute these commands, they can be sent to the command line. PEEK will also display the value of the memory word stored at that location and the one following in hex and decimal.

Parameters can now be passed to batch files. In the batch file itself, these are referred to as %1 through %9. With this, you can create general purpose batch files to automate tasks.

Internally, SDX is very different from earlier versions of SpartaDOS. All of the files on the cartridge are relocatable and can be held in memory. COMMAND.COM, the command processor is one of these files. It is non-resident in nature and is unLOADed from memory when binary files are run. This saves about 4K of memory. It is reLOADed when the program is exited to DOS. Disk based programs written in relocatable format could be loaded at MEMLO, and held, eliminating the need to reload from disk each time. Unfortunately, information on how to

THE SPARTADOS X cont.

write these modules is almost non-existant, so for now, only the programs on the cartridge can be held.

Some of the new commands and changes with SDX not mentioned above:

CHTD/CHVOL - now built in.

COPY - now checks to see that there are two files specified. Files could be lost with disk based SpartaDOS by accidentally not specifying a second filename.

DIR - /p directive pages output, /c directive gives file count.

DUMP - Hex dump of file.

FIND - search all drives for filename.

MEM - displays banks available, extended memory.

PATH - Set search path.

PROMPT - set system prompt with meta-strings.

RS232 - now built in.

SET - display/set environment variables.

UNERASE - restore file(s).

X - load file/disable cartridge (for long binary files).

This is an incomplete listing of the features of SpartaDOS X. There are many others and new uses for the functions appear constantly. While learning to use SDX will take some time, it is well worth it in the end.

REPAIRING THOSE FUNCTION KEYS

by Dean Lowery

One of the most common faults with the Atari 65XE and 130XE is the keyboard , and in particular the function keys which are usually the first go.

Why I hear you say, well because the key tracks on the actual

FUNCTION KEYS cont

keyboard oxidise and the signal can't get through. One solution is to clean the tracks but you must be careful not to wipe the tracks out, another is to wire push buttons directly to the keyboard socket inside the computer.

By the way I am typing this article in on my Atari 800XL which is 9 years old and I haven't had any problems with the keyboard yet!

The keyboard in your Atari 65XE or 130XE is connected to a 24 pin socket which carries all signals from the keyboard and then to computer which figures out which key has been pressed.

Looking at the socket from the solder side the first 4 pins on the left carry the signals from the function keys, (see diagram 1) so if one of the keys is broken then you can wire up a push button here, which is very simple and anyone who has the ability to solder can do it.

WIRING UP

For this modification you will need:

Mini SPST Momentary Pushbutton Switch, however, if many function keys are broken they can be bought at Tandy's at 1.99 for 4 switches.

2 core wire
Pair of wire cutters,
Pair of long nosed pliers,
Soldering Iron,
Solder and a screwdriver.

Turn over your machine and undo the 4 screws holding it together, and remove the casing. Now gently ease the keyboard connector from it's socket and put it to one side.

Now you will need to remove the shielding, to do this use the long nosed pliers to straighten out the metal hooks holding the shielding on, there are 7 of these. Once you have done this remove the shielding and put it to one side with the keyboard and the casing. You should be down to the circuit board.

Now remove the 7 screws holding the circuit board to the bottom half

REPAIRING THOSE FUNCTION KEYS continued

of the casing and put them to one side.

Gently lift the circuit board from the casing and you will notice another layer of shielding on the bottom of the circuit board this should pull off easily, put it to one side.

As you can see from diagram 1:

Pin 1 is for RESET
Pin 2 is for OPTION
Pin 3 is for SELECT and
Pin 4 is for START.

Cut off 2 lengths of wire, strip both ends and of each wire and tin them with solder.

Now solder one of the wires to a leg of the pushbutton and the other wire to the other leg of the pushbutton.

Right so far so good, now you want to wire up the switch to the pin of the function key that is broken. For example if your RESET button was broke solder any wire to pin 1 or if your SELECT button was broke solder any wire to pin 3. Easy!

Do not keep the soldering iron on the circuit board for too long as electronic components can be damaged if too much heat is applied.

Solder up one wire of each switch to as many keys that are broken, the other wire is the ground wire and will be dealt with later.

Now replace the bottom half of the shielding and run the wires out somewhere making sure that none of the wires are snagged. Replace the circuit board back in the bottom half of the casing and replace the screws. Now to deal with the ground wire.

If you look at the side of the circuit board with the components on, you will see a silver strip running around the outside, the screw holes are in this silver track so all you have to do is wrap the ground wires you have to any screw and replace it. Nearly finished.

Now to place the pushbuttons. You can run them out by the joystick sockets or if you want you can drill holes in the top of the case for the pushbuttons and screw them into there.

Oh, by the way put your machine back together.

TESTING YOUR NEW KEYS

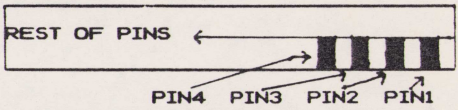
Turn on your machine and go into BASIC. Type BYE to access the self test. Select the keyboard test and test your new keys. If they don't work turn off the machine and check your work. However all should work fine.

NOTICE

If you want to do this modification do it at your own risk. If you damage your machine (which is highly unlikely) then TWAUG or myself will not take any responsibility. If in doubt ask someone who knows what they are doing or search for a new keyboard.

DIAGRAM 1

KEYBOARD CONNECTOR 24 PINS



VIEWED FROM COMPONENT SIDE

LACE

The London Atari Computer Enthusiasts.

As a member of LACE you receive a monthly newsletter and have access to a monthly meeting. They also support the ST with a good selection of PD sotware.

The membership is £8.00 annually.

Write to the secretary of LACE for more information:

Mr.Roger Lacey
LACE Secretary
41 Henryson Road
Crofton Park
London SE4 1HL

GAME REVIEWS

by Kevin Cooke

Well, it's good to be back again for some more reviews. At present I haven't had any comments or suggestions about my column so I'll assume that most people are happy with my reviews as they are. Don't forget that you can write to me, c/o TWAUG with any comments or suggestions. So, without further ado, here are this issue's reviews. I've tried to include a game to suit every pocket so read the reviews and take your pick. Look out for my "games in the pipeline" feature after the reviews!

Title: T-34 THE BATTLE
(130XE required)
Sold at: Micro Discount (Derek Fern)
Price: £5.95

Well, what can I say? From the moment the disk is booted, the game oozes class. The disks starts with a pretty clever demo. A jungle (viewed from above) comes into view, accompanied with digitised sounds of birds, etc. the screen slowly starts to scroll downwards until a tank comes into view. The tank searches for another tank on the horizon before letting rip with it's guns, again accompanied by digitized sounds. Finally, a digitised explosion is heard and the screen fades out. Last but not least, the words "T34" and "THE BATTLE" come crashing into each other on metal sheets, just as is seen in the Terminator film. Again, digitized sounds accompany this. Now, I'm not saying that an excellent intro demo means that the game will be perfect but this really got me excited.

After flipping the disk, the game itself loads. This is quite a long process but is understandable for a game using digitised sound.

The first thing you get to see is the title screen. This is fairly colourful, showing all of the options available. Wind speed and direction can be altered, along with the type of landscape in your tank's way, the gravity present, and also various "objects". These options can be changed by the use of a joystick controlled pointer.

When you decide to start the game, the screen goes worryingly blank for a few seconds. However, the main screen does appear, split roughly in halve. At the bottom of the screen in your tank and a second player's tank. Each player has to aim their gun so that the shell which is fired lands on their opponent's tank. However, your shots are hindered by the fact that there are various mountains or hills in the way, just waiting to protect your opponent! The only way to make sure you hit your opponent is to change the power you put into your shot and the angle at which you fire it. Both of these changes are made via "click-on" icons at the top of the screen. It is also possible to move your tank slightly backwards or forwards and scroll the screen so that you can actually view all of the mountains in your way. When you do finally decide to shoot a shell, the screen is completely taken up with the main game screen and, after a slightly muffled digitised yell (saying "ready... commence firing?"), you get to see your shell go flying through the air, usually missing your opponent! Each player takes it in turn to shoot until one player hits the other.

"But what is the game like?" I hear you asking. Well, quite simply brilliant! The addictive

GAME REVIEWS continued

quality is huge and you just can't help having "just one more shot"! Even the comical touches (such as the various things which happen to your tank when hit) add to the enjoyment of the game. The digitised sounds also add a nice bonus and the option to change wind strength and direction ensure the game's difficulty level can be increased. Frankly, if you enjoyed Megablast, you'll probably love this one just as much! I wouldn't have minded a bit of music while you are preparing your shot but, even so, the game is great fun. Don't forget that this game will only run on a 130XE or similarly upgraded.

So, overall there is only one problem with this game - there aren't more like it!

Title: NINJA
Sold at: Page 6
Price: 0.95p

Before I start this review, I must just say one thing which new Atari 8-bit users may not know. Several years ago, budget software on cassette tape was all the rage. Some of the titles released were excellent, some were good and a few were awful! However, generally the standard was acceptable for the modest price of around 2 pounds 99p. Just lately Page 6 have taken stock of several of these titles, Ninja being one.

I case you haven't already guessed, Ninja is a martial arts game and one of the better ones at that. The storyline says that a princess has been kidnapped and you are the only one that could rescue her - yes, I know we've all heard the storyline about 100 times before in completely different games! The game starts off with your character standing in an outdoor scene overlooking the sea. Below this character are two ninja throwing stars (shuriken) and a knife which he can pick up and throw at enemies. The ninja also has a range of attacks which he can perform on his opponents, including the devestating attack with his sword.

The main game is made up of many different screens, each containing at least one enemy. Whenever you enter a new room, the screen goes blank and the name of the room is announced on-screen before it actually appears. The object of the game is to find six idols which these enemies are guarding and to collect them to reveal the secret room where she is being held.

The graphics throughout the game are VERY impressive, especially for a game selling at a price this low. Your character and those which you have to fight are quite large and very well defined - their movements are also very realistic. Difficulty wise, the game seems to offer a lasting challenge. I have completed it and can safely say that the ending is not particularly exciting but, then again, this complaint can be made to almost any other piece of Atari 8-bit software. The only thing which lets the game down is the instructions. These do not tell you that pushing up on the joystick when underneath a hole in the ceiling will make the ninja jump up one level. This piece of information is, in fact, a very important thing to know if you want to advance in the game.

GAME REVIEWS continued

Overall, ninja really impressed me when I first bought it and still holds my interest some 5 years later - beat that super turbo hyper fighter VII!

Well, that's my complete reviews over for this issue but what about titles which are "in the pipeline"? Well, Derek Fern has recently sent me five impressive looking games. Please note that he currently has no instructins so this is only what I could work out for myself.

First up is BARBARIAN, based on an old ST/AMIGA game called..... well, if you can't work that out for yourself, lets just say that the name wasn't any different! The game is a one or two player fighting game between two barbarians, each trained in a number of violent attacks. How about cutting your opponent's head off?!! The game seems to be VERY hard against the computer but better in one player mode. This one could be good.

TANKS is a one or two player game in which you can take control of, strangely enough, a tank! The game advances to new levels after you have shot lots of the computer controlled tanks.

TRON is based on the light cycle sequence from the film of the same name. This certainly appears to be the best version of this so far.

DEATHLAND is a platform and ladders type game in which the object is to rescue hostages who have been tied up. The idea of this seems slightly similar to Aliens on other consoles. Very good fun.

TECHNUS is, again, a platform and ladders type game but this time seeming slightly stranger to play. Graphics are good and there is an excellent digitised tune on title screen. Another to look out for.

I also know that Derek is hoping to get a program that will convert any standard 62 sector picture file into a "magic eye" or "stereogram" type picture which can then be printed out.

Also in the pipeline are CAVEMAN, TOP SECRET, AROUND THE PLANET, INNY SWAT, NAJEMNIK I & II, STARBALL, TEACHERS KILLER, GLOBAL WAR, WYPRAWY KUPCA and finally, TERMINATOR. What is also great is that some of the above named titles can serve as tongue twisters if repeated several times! Brilliant!!

OK, I think I've taken up enough space for this issue! Don't forget, you can write to me (c/o TWAUG) if there are any particular pieces of software which you would like reviewing or any suggestions you have about my column. Would you like longer and more detailed reviews, more shorter reviews, no reviews?!! Let me know and I'll see what I can do. Enjoy the rest of the issue.

TWAUG
8-BIT
T.W.A.U.G.
P.O.BOX No.8
WALLSEND
TYNE & WEAR
NE28 6DQ

THE PRC CONSTRUCTION SET

By Bill Walraven

This is a set of tested Turbobasic PRoCedures. They are used as the muscles (or building blocks) of a program, which consist of a skeleton structure that calls the PROC's, at the appropriate time & place.

1 - A building block is a PROC or a set of PROC's.
2 - It is situated between fixed line numbers. E.g. PROC LOAD runs from lines 200-300.
3 - Its variables, strings or arrays, must not clash with others.
4 - The dimensioning of above is done in PROC INIT, which runs from 21000-21200. Each proc has one or more lines allotted to it within PROC INIT.
5 - Each block, -! if dimensioning is needed! - contains the line,
-> 20 EXEC INIT <-

Successive PROCS entered one after another, will not interfere, but only substitute their own 'line 20', by overwriting the existing one.

I use the toolbox when I make a program & need to use a SUBROUTINEx.

The PROC needed is just entered and inserts its lines between those of the program.

This leads to the next step.

Suppose I have various finished programs on a disk, using PROC'S. They may have some PROC's in common. So on my disk, the same PROCS may be present a number of times, be it in different programs.

So I thought, can I save programs without these PROCS and incorporate the necessary PROC's at the time the program runs?

My first effort was a program with a line "Please enter the following listed programs first, then RUN again". So one entry after another was necessary, with a READY after each entry, and finally the command RUN.

Laborious, extra checking's needed, etc. and Murphy likes that.
Alternatively I could use 'dynamic entry', but that too is cumbersome and less elegant than the solution I use now.

This is to convert the PROC'S into SPR'S -> Self entering ProC's <-

This does the trick.
- enter each line and store it in a string, complete with final RETURN (155).
- after the last line is entered add "GOTO 10" + RETURN, to the file.

On entry, each line will be absorbed into the program, (n.b. if a line number equals an existing one, that one is replaced!!).

The GOTO 10 + CHR$ (155), will be seen as a direct command and the program carries on, no operator needed to enter another listing, or typing RUN.

IMPORTANT!! programs entering SPR's must (if following the last example) have a LINE 10 to go to. E.g. have a line -> 10 REM <-

So I saved on duplicate

THE PRC CONSTRUCTION SET continued

subroutines? Is a further savings possible? Yes!
When we type 'GR.8', our ATARI will expand this all by itself to -> 'GRAPHICS 8' <-

It extends the abbreviation and adds spaces (before the 8) where needed in a basic line.

BUT IT MUST NOT DO SO IN A TEXT LINE, OR AFTER A REM!

I wrote a short program (CONVERT), that converts a PRC to an preshrunk SPR and saves it as such.

I add this program (COMPACTG .OTO, as it compacts & ads a GOTO) that shows also the number of bytes saved, and their place in the listing as a demo of its activity.

Savings are about 15%.

I welcome all suggestions and subroutines, for inclusion in such a library. I feel that if we make a TWAUG TOOLBOX, we must coordinate all routines entered, to avoid clashing variables & line numbers.

The programs I send are in the nature of suggestions/demo's, presently a BAT file as submitted by me, includes the lines 4, 10, 15 & 17,

```
e.g. 4 DIM(SPR$)
     10 REM
     15 READ SPR$: IF SPR$<>
"FINISH" THEN ENTER SPR$
     20 DATA D:CSORT.SPR,D:SECT
MOVE.SPR,FINISH
```

I've added the following programs:

DEMO1SPR.TBB & DEMO2SPR.TBB: 1 will load 2 showing the working of SPR'S

GR0TOGR8.BAT: demo moving a GR.0 screen to GR.8. This is useful for printer dumps (I can do a GR.8 dump on the SEIKOSHA GP100, that I can't do otherwise)

DIRALF.BAT: gets directory, alphabetizes it and shows it on screen. No use here of open# 0,6,0,. Sectors are read, so file data may be translated and shown.

APOLOGY FROM MAX

In issue #13 I published an unfinished article by the headline of "SEARCHING FOR BOBBY FISHER ON THE 8-BIT ATARI", and I also credited it to the wrong person. If you check issue #13 page 18 you will see that it is credited to Bill Hall, it should have been in fact ED.HALL. I am very sorry about this grave error, so please accept my apology.
And the unfinished article ends and I quote:

"Chess"

"Rather surprisingly, this program by John Krause is the only Basic chess program for Atari 8-bit."

Neither David, who made a copy from the original disk he received from Ed Hall, nor I, have any idea what happened to the rest of the article missing of that disk. As soon as David informed me that I published only part of the article, I immediately checked the disk with an editor and found

APOLOGY cont.

that the end part was missing completely. So please accept my apology for this mistake also and find the rest of that article at the end of this apology. It is a complete mistery to us but I hope this hasn't spoiled your enjoyment to much.

SEARCHING FOR BOBBY FISCHER ON THE 8-BIT ATARI:

An Annotated List of Available Programs

By Ed Hall.

Chess

Rather surprisingly, this program by John Krause is the only BASIC chess program for Atari 8-bits. It appeared as a type-in listing in the December 1984 issue of COMPUTE! Unfortunately it does not check for legal moves. Nice graphics. Joystick only.

Master Chess

A budget offering from Mastertronic. Decent graphics but few features. Algebraic notation.

Mychess II

By David Kittinger, Atari version by Walter Hochbreuckner (1984). This under-rated version from Datamost has lots of features and a good manual. It also made the first serious attempt to make graphics an important feature; unfortunately the colour scheme nullified its efforts in this area. Mychess II was also the first game with 3D graphics, and

SEARCHING FOR BOBBY FISHER ON THE 8-BIT ATARI: continued

although the game is virtually unplayable in this mode, it's still a neat hack. Algebraic notation.

Chessmaster 2000

This is the Cadillac of Atari chess programs. It has tons of features, a good manual, and gorgeous graphics. Though no credit is given for the game's development, its similarities to Mychess II are too great to be a coincidence. Movement by joystick or algebraic notation. The Software Toolworks (1986).

Colossus

A superb program by Martin Bryant. Many features, with crisp and pleasing graphics. Movement is by cursor keys or algebraic notation. Surprisingly, this is the only Atari 8-bit version with a clock.

Colossus 3.0 (1983, 1984) was marketed in North America by ANTIC. It was followed by Colossus 4.0 (1987) and an ST version, Colossus X. In issue 37 of Page 6 magazine, John Davison pitted Colossus 4.0 against Colossus X. In 12 matches, both programs had identical records of 5 wins, 5 losses, and 1 tie. The docs for Colossus 4.0 report the results of 16-game matches against the following programs:

W	L	vs
16	0	Master Chess
16	0	Atari Chess
16	0	Parker Chess

SEARCHING FOR BOBBY FISHER ON THE 8-BIT ATARI continued

16 0 Odesta Chess
13 3 Sargon 3
11 5 Mychess II

SuperQuerg Chess

The newest chess program for the Atari 8-bit appeared as a disk bonus in New Atari User issue 64 (October/November 1993). Written by John White, this program does not have as many bells and whistles as Chessmaster 2000, but it has all the essentials and plays a strong game. It is also "unusual among chess programs in that it relies more on the strength of its positional play than on its tactical play." Simple but pleasing graphics. Algebraic notation.

Note: John White is the author of "Writing Strategy Games on Your Atari Computer" (Sunshine Books, 1983). It includes discussions of various chess programming techniques and some short BASIC programs. John White also wrote NegaQuerg II, a suicide chess game where the object is to lose all your pieces. It was part of the New Atari User disk bonus mentioned above.

1200XL/800XL 256K MEMORY UPGRADE

By Bob Woolley

If you don't have a reasonable amount of experience in soldering

MEMORY UPGRADE cont.

and electronic construction, get some help before you try this upgrade... I had an IC upgrade go bad when I converted one of my 1200s (U25 - not because of the upgrade). If this were to happen to you, could you get it fixed? know someone with a scope? if not, think about it!!!

Still game? OK..Take your computer apart and remove the shields. If you are careful, you could fit this under the shields, I suppose, but you are already off to a bad start. Take some time to familiarize yourself to what is going to happen here... play a little Star Raiders while it's still working. A silent prayer to the Fuji god maybe in order.....

Build the circuit on a small grid board sockets and small diameter wire (wire-wrap wire is perfect, comes in .5in. increments from .5in. to 6.0in and it is already stripped). It is best to mount a socket for the 16 pin jumper, although I built one of these on a small board and wired the jumper directly to my board - it's a mess to troubleshoot. The best method is to use sockets for both cables and BE NEAT...it only took me three hours to wire up the board, so take your time...you won't save time if you make any mistakes.

Most 1200XLs have 24 pin ROMs in U12 and U13... they must go!! Desolder the sockets CAREFULLY and solder in GOOD QUALITY 28 pin sockets. Use a good desoldering tool to remove the

1200/800XL MEMORY UPGRADE continued

old sockets - you can't do it any other way - bet your 120XL on it?? To the right of the ROMS are a column of white bodied jumpers (0 ohm resistors). Remove W7, W8 and W9 and (using wire) jumper W11, W12 and W13. Get (burn 'em on 2764's if you can) 2 800XL OS ROMs and plug these into your new 28 pin sockets. If your 1200 already has 28 pin sockets, just plug in the chips.

Remove all 8 RAM chips, and replace them with 21256-150 RAMs. Notice that I used CMOS ICs where they are available. If you can't get CMOS (74HCTxxx is CMOS), you can substitute an LS version (74LSxxx). DON'T do this for all of the chips... the CMOS version draw very little power from your Atari; you MAY overheat your power supply if you use all LS parts.

Trace pin three on each 74LS158 (U10 and U7) to the underside of the board where you can swap those two pins. Cut the foil and wire from one to the other... you want pin three to be swapped between the two ICs...

Use SHORT lengths of ribbon cable to connect to the board - small diameter stuff is easier to solder in. The cable to U10 (U27 in 800XL) should terminate in a 16 pin DIP plug that plugs into the existing socket. The other ribbon cable (from REF, HLT, SC etc.) will solder directly to the bottom of the board - right at the IC pin called out in the schematic notes.

I would measure between +5V and ground with an ohmeter for 100 ohms before I power up the board ... I had them shorted at first!!!

Now you have a 130XE type machine. 16K banks of RAM can be selected by bits 2,3,6 and 7. The 130XE software only uses bits 2 and 3 - bits 6 and 7 are another 128K of space. PLEASE NOTE: The NORMAL 4K block will be selected if bits 6 and 7 equal 00 - even if bits 4 and 5 are selecting the extra blocks. Remember, you only have 256K here, not 64K + 256K.

If you want absolute 130XE compatability, connect S0 and S1 to +5V - you now have only 128K -- but you don't have to worry about some program that trashes bit 6 or 7 in $D301. As an added bonus, you will notice the two extra leds on your machine, L1 and L2, are now indicators, of the two extra memory selection bits -- blinking lights and all... A REAL computer now!!!

Oh Yes, you 800XL folks, I haven't tried this on an 800XL but it should be OK... Maybe someone could let me know?? You do not need to do all this extra pin three swapping or ROM replacement... but then again, you don't have much room or any function keys (F1-F4).

DIAGRAM: on next page

1200XL/800XL MEMORY UPGRADE continued

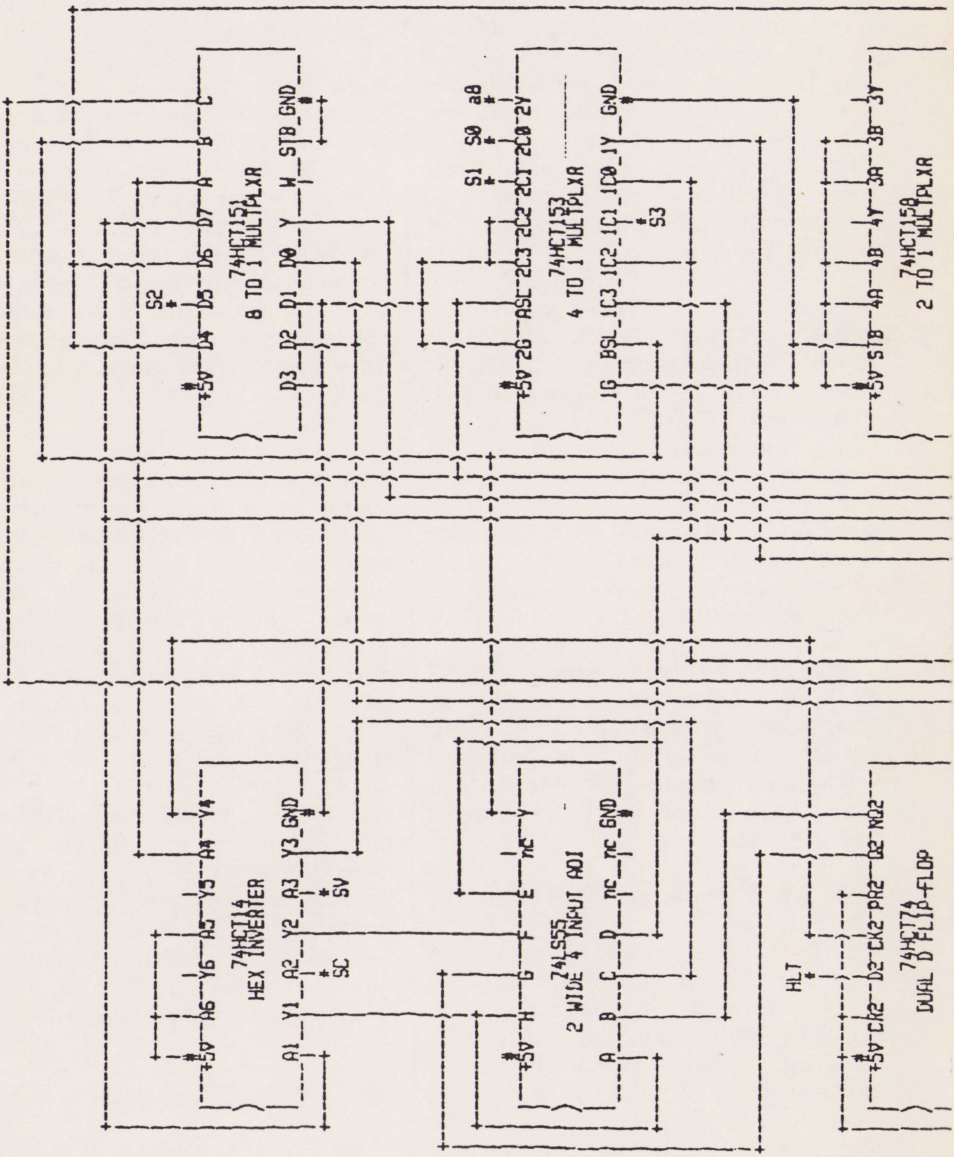
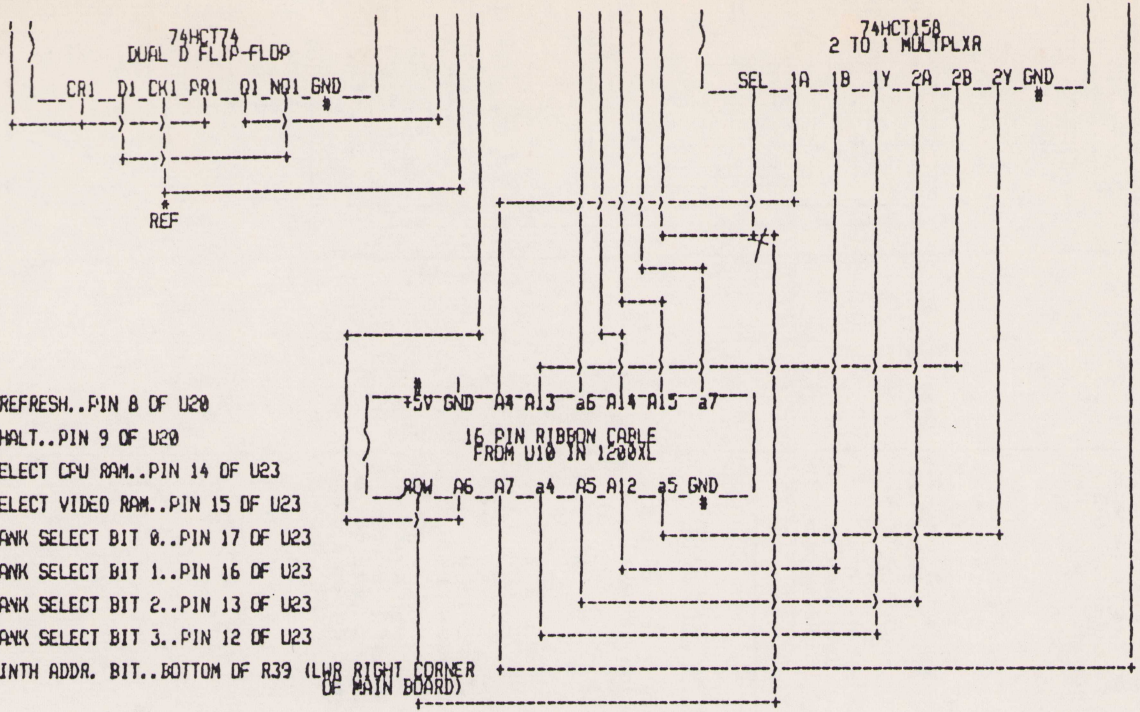


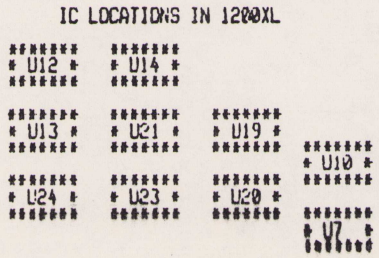
FIGURE
256K memory expansion

1200XL/800XL MEMORY UPGRADE continued



*REF - REFRESH..PIN 8 OF U20
*
*HLT - HALT..PIN 9 OF U20
*
*SC - SELECT CPU RAM..PIN 14 OF U23
*
*SV - SELECT VIDEO RAM..PIN 15 OF U23
*
*S0 - BANK SELECT BIT 0..PIN 17 OF U23
*
*S1 - BANK SELECT BIT 1..PIN 16 OF U23
*
*S2 - BANK SELECT BIT 2..PIN 13 OF U23
*
*S3 - BANK SELECT BIT 3..PIN 12 OF U23
*
*a8 - NINTH ADDR. BIT..BOTTOM OF R39 (LWR RIGHT CORNER OF MAIN BOARD)

IC LOCATIONS IN 1200XL



ALL *+5V PINS ARE CONNECTED TOGETHER ALTHOUGH NOT SHOWN IN DRAWING.

ALL *GND PINS ARE CONNECTED TOGETHER ALTHOUGH NOT SHOWN IN DRAWING.

PIN 1 ON ALL IC'S IS IN LOWER LEFT HAND CORNER ... PIN 8 IN LOWER RIGHT (TOP VIEW)

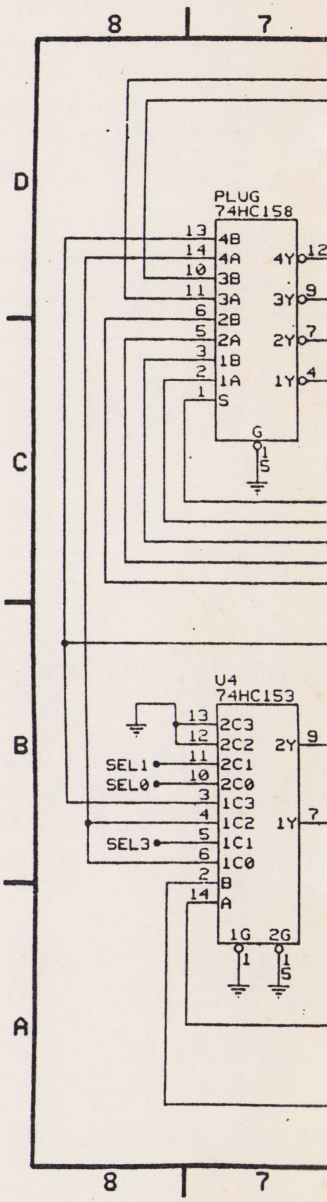
) SHOWS CROSSING WIRE ... NO CONNECTION

+ SHOWS CONNECTED WIRES

800XL UPGRADE

This upgrade is designed to fit in an 800XL and attach to the motherboard through both discrete wires and a 16 pin DIP flat cable. This cable connection is designated as PLUG on the schematic and it replaces the U27 chip. The other wires are shown as bullets and attach at the following points.

* REF - refresh — pin 8 of U7
* HALT - halt — pin 9 of U7
* CSEL - CPU select — pin 14 of U23
* VSEL - ANTIC select — pin 15 of U23
* SEL0 - bank bit 0 — pin 17 of U23
* SEL1 - bank bit 1 — pin 16 of U23
* SEL2 - bank bit 2 — pin 13 of U23
* SEL3 - bank bit 3 — pin 12 of U23
* A8 - ninth address bit — right side of R32 (on upper left corner of main)



TWAUG ANNOUNCEMENT

TWAUG now have a good supply of blank disks for sale. Prices include P&).

QUANTITY PRICE

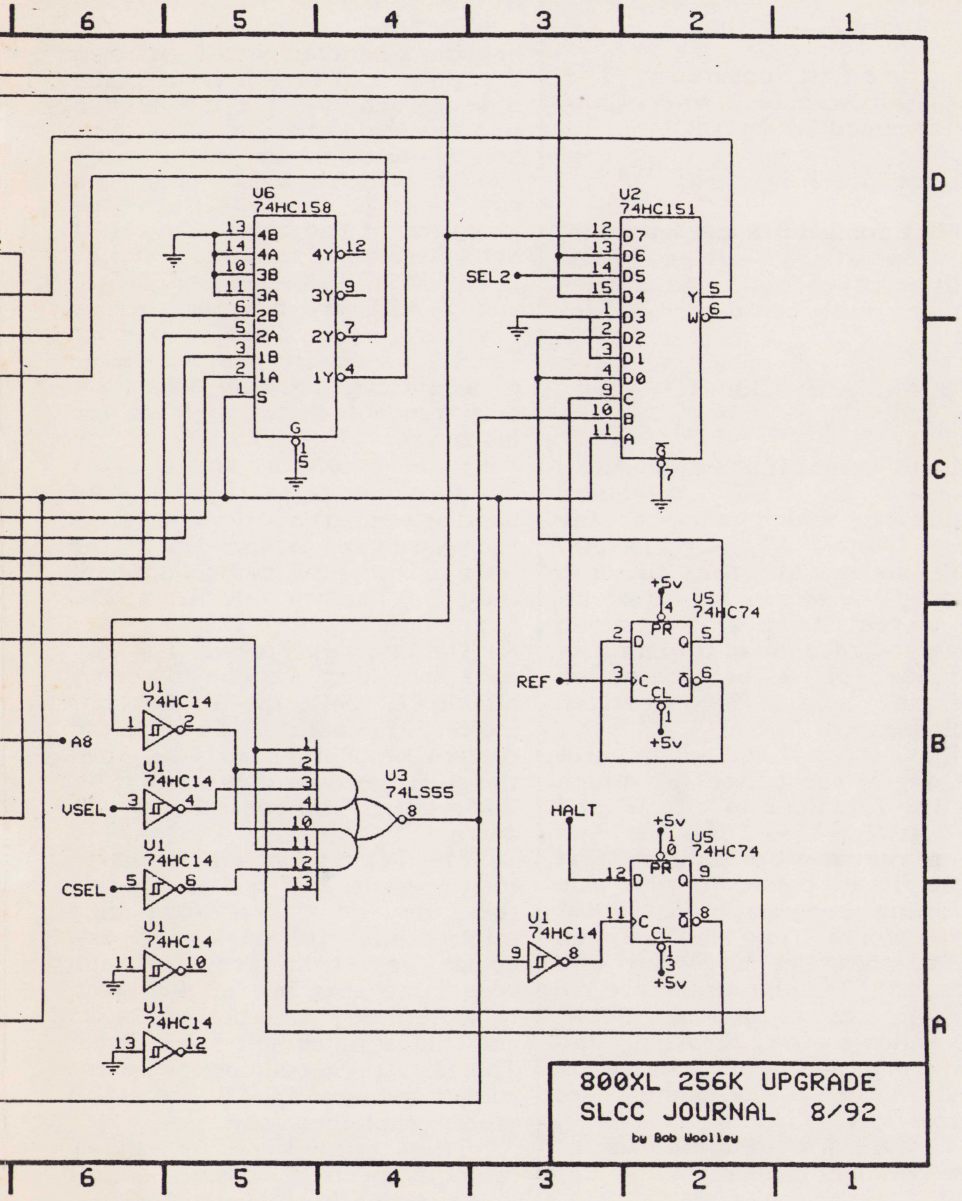
10.....£1.50
25.....£3.50
50.....£6.00
100.....£11.50

Do you have an XF551 drive? if so, we can supply you with good quality double sided disks. These disks are now very hard to get hold of, so if you wish to get some while our stocks last, get in touch as soon as you can. Prices include P&P.

QUANTITY PRICE

10.....£2.50
25.....£5.50
50.....£9.00
100.....£17.00

800XL UPGRADE



CRACKING THE CODE

by Keith Mayhew

Re-printed by M. Gerum

This article first appeared in "The UK ATARI Computer Owners Club" later renamed "MONITOR"

Part 16

Continuing with our study of the operating system, the facilities of each of the system handlers will be described in-depth.

KEYBOARD AND SCREEN HANDLERS

Input from the keyboard is provided by the keyboard handler, 'K:', which converts key presses into ATASCII (ATARI ASCII) codes. All the standard modes of the screen are set up by the screen handler 'S:', which allows the reading and writing of individual pixels or characters, depending on the selected graphics mode.

The third handler in this category is the editor, 'E', which uses the functions of the 'S:' and 'K:' handlers to provide an interactive way of editing text before it is passed on to the requesting program. The editor only works in the screen handler's graphics mode zero and forms the familiar interface to many programs, such as BASIC. This allows the deletion and insertion of individual characters or lines in a consistent, yet flexible way.

We will now examine each of these handlers in turn.

THE KEYBOARD HANDLER K:

The main function of the keyboard handler is to wait for a key press and return its ATASCII code. To achieve this, it works in two separate halves. The first halve consists of an IRQ interrupt handler which is called each time any key is pressed, with the exception of the control and shift keys. Remember that the function keys 'RESET', 'OPTION', 'SELECT' and 'START' are not part of the keyboard handler's domain; 'RESET' is specially handled via its own interrupt and the others are only readable directly from the hardware.

Upon receiving an interrupt due to a key press, the keyboard handler reads the key's code from the hardware along with the state of the shift control keys at the time. The variable 'CH' at 2FC hex is used to store the key code for the last key pressed. The key code itself occupies the lower six bits of 'CH' while the top two bits indicate the state of the shift and control keys; if bit 6 is set then the shift key was down and if bit 7 is set then the control key was down.

The Table 1 lists all the sixty-four possible key codes, ignoring the top two bits of shift information, followed by the actual key that generates the code. Note that not all keys can be generated from the keyboard and this is indicated by 'Not Used'. The last three columns represent corresponding ATASCII codes and will be explained soon.

The interrupt handler responds specially to the control-1 key combination by not storing a

CRACKING THE CODE continued

key code in 'CH' but toggling the state of the 'start-stop' flag variable. 'SSFLAG', at 2FF hex, between its usual value of zero and FF hex. 'SSFLAG' can be monitored by other routines to halt their output whenever it is set to FF hex; both the screen handler and the editor are affected by this flag and enter a loop whenever they see it active.

A zero is also stored in the variable 'ATRACT', at 4D hex, whenever a valid key code is written to 'CH'. This resets the attract timer, which, if allowed to count up to over 127, starts the cycling of the colours on the screen to attract attention because no one has pressed a key for about ten minutes! If you write a program which may not make use of the keyboard then you might consider periodically writing a zero to 'ATRACT' to prohibit colour cycling. Alternatively, you could disable the VBI interrupt routine which implements the counting and colour cycling.

The 'break' key is also handled specially, in that no code is written to 'CH' but zero is stored in the following variables: 'BRKKEY', 'SSFLAG', 'ATRACT' and 'CRSINH', 'BRKKEY', at 11 hex, is a flag which indicates to other routines that the break key has been pressed. This is used by most handlers to indicate user-abort and generates the appropriate CIO error code for a break; after a routine detects the break key flag set to zero it should set it back to its normal value of 80 hex to clear it. Pressing break also re-enables screen output if it was stopped; resets the attract timer; and re-enables the cursor of the editor handler if it was previously inhibited, i.e. hidden by setting 'CRSINH', at 2F0 hex, to a non-zero value.

The other half of the keyboard handler is activated by CIO in response to a 'get key' operation. It examines the variable 'CH' and waits until a value other than FF hex is found, i.e. a key was pressed. It then resets 'CH' to FF hex ready for the next key and processes the key code it obtained. If the key code is valid an audible click is generated, either through the keyboard speaker for the 400 or 800 models, or via the sound output for the XL and XE. If a key is held down then no further interrupts are generated but key repeat is implemented in part of the VBI code by simply storing the key code into 'CH' at regular intervals for as long as the key remains down.

A look-up table is used internally to translate from the key codes to ATASCII codes which are then returned to CIO. Table 1 shows the associated ATASCII codes in the last three columns; the first is for an un-shifted key, i.e. without shift or control; the second is for the same key but with the shift key; and the third is with the control key. Note that not all shift and control combinations return a value at all, in particular, any key combination where both the shift and control keys are held down simultaneously are ignored.

There are two variables which can affect the values returned from the keyboard

CRACKING THE CODE continued

handler: 'INVFLG' at 2B6 hex and 'SHFLOK' at 2BE hex. 'INVFLG' is the 'inverse' flag which is toggled whenever the ATARI key (half shaded box on XL/XE) is pressed. Normally this flag is zero, but when set to 80 hex it causes further returned values to have their top bit set, the following key combinations are the exceptions: escape, cursor-left/right/up/down, clear-screen, back-space, tab/set/clear, return, control-2, insert/delate-character/line.

The shift-lock variable 'SHFLOK' can have one of three values: zero is normal mode; 40 hex is shift-lock, and 80 hex is control-lock. The 'caps-lower' key when pressed does not return a value from the keyboard handler but is used to change the state of the 'SHFLOK' variable. The 'caps-lower' key by itself will set the normal mode where all further 'letter' keys will return the codes for lower-case letters, i.e. un-shifted. When 'caps-lower' is pressed with shift all further letters are returned as upper-case. When 'caps-lower' is pressed with the control key all further letters are returned as if they were typed with the control key down. Note that no matter what lock is currently selected pressing either the shift or control key with a letter key will over-ride the lock.

The keyboard handler responds to the following CIO commands:

OPEN:
Device name is 'K:'; read only.

CLOSE:
No action taken, just releases IOCB.

GET CHARACTER:
Reads a single key and returns its ATASCII code, waits if necessary.

GET RECORD
Reads keys until return is pressed.

GET STATUS
No action taken, status set to 1 for OK.

The following error codes can be returned from a get character or record operation:

80 hex: Break-key pressed.
88 hex: End-of-file.

Both of these errors conditions return the end-of-line code 9B hex, i.e. as if return was pressed. The end-of-file error is generated by control-3 and so, for example, you could copy from the keyboard device to another, say a disk file, and terminate the transfer with control-3.

You may find that having the get character operation wait for a key particularly awkward, in which case you can test the variable 'CH' yourself for a value other than FF hex before getting the character.

THE SCREEN HANDLER S:

The screen handler can set up a screen display in any one of sixteen pre-defined formats and provides the ability to read or write any character or pixel of the screen. Due to its general

CRACKING THE CODE continued

purpose nature, screen I/O is no where near as fast as it could be if you accessed the data directly, but there is no reason why you cannot mix access between CIO and direct manipulation to obtain the best of both worlds.

Once the 'S:' device has been opened a display list is set up and the display area is initialised. At this stage you may decide to modify part of the display list to suit your application. The screen handler may still be used but it will not know of your modifications; as long as you are careful, you can use the screen handler effectively to access most parts of your screen, usually the text segments and update the rest by direct access.

Table 2 lists the sixteen modes which the 'S:' supports and shows the corresponding ANTIC mode number of the lines which constitute the basis of the display list. The number of lines down the screen are shown for both a full display and a 'split' screen, if it is available. A split screen mode has four lines of text mode 0 appended to the end of a standard screen.

All the display lists built by 'S:' have three eight-blank-line instructions at the top to ensure the main display will be visible on all displays. Note that modes 9 through 11 are the same as mode 8 except that they turn on one of the three special GTIA colour modes and that modes 12 through 15 are only available with the XL/XE operation system. The following descriptions will assume you know the individual characteristics of the graphics modes, if you do not, you might wish to refer to parts ten, eleven and twelve of this series.

To open the screen device you need to specify the device name 'S:', the type of access is required in auxiliary byte 1 and the graphics mode in auxiliary byte 2. The access type consists of the usual read/write selection plus two further bits: bit 5, if set, means that the screen will not be cleared; bit 4, if set, causes a split screen mode to be built-if it is not supported for the specified mode then a full screen is built instead. Note that both of these options are ignored if if you speciy mode 0 in auxiliary byte 2.

The split screen mode is only available if you have previously opened the editor device, as it is required to control the text portion of the screen. As the editor is opened by the system on IOCB zero, you will usually be able to open 'S:' straight away in split mode, however, it is safer to close IOCB zero and re-open it for the editor device to ensure that it is open.

Assuming the the open was successful, then several internal variables are initialised. The current cursor's row and column positions are both set to zero, which corresponds to the top-left of the screen. The row is held in 'ROWCRS' at 54 hex and the column is held in two bytes starting 55 hex in low/high format. The 'cursor' never really exists on the screen in any mode other than zero, but the position determines where the next put or get operation will take effect. In mode zero the cursor's position is shown by inverting the character

CRACKING THE CODE continued

at the specified point. If you wish to turn off you can set 'CRSINGH' to any non-zero value, as already explained, but you must follow this by some screen operation, such as getting a character, to have it actually turned off, alternatively, you could invert the character at the cursor position yourself by directly clearing its top bit.

The address of the screen data start location is stored in low/high format at 'SAVMSC' starting at 58 hex: you can use this pointer to gain access to the screen data yourself, or you can changeit and have 'S:' read and write a totally different area of memory, for instance a second screen. If you have modified the display list and wish to access a portion of the screen which is out of the standard range of the cursor, you can move 'SAVMSC' to point to the new area, effectively moving the origin of the screen.

Although opening 'S:' sets it up for one particular mode, it is very easy to make it work as if it were in a different mode y altering the mode number in the variable 'DINDEX', at 57 hex. with a combination of this method and altering 'SAVMSC', as above, you can easily deal with mixed mode display lists.

The put character operation places the specified data at the cursor position and moves the cursor to the right by one pixel or character. In text modes, all eight bits of each data byte are used but this is translated into the appropriate character code before it is stored in screen memory. For graphics modes only one, two or four of the low order bits are used, depending on the **mode, and** these are stored directly into the corresponding bits in the display area.

In all modes, the end-of-the line character, EOL (9B hex), does not put any dat on the screen but moves the cursor's position to the start of the next line down. Similarly, the clear screen character, 7D hex, will always clear the entire screen and set the cursor back to the top-left 'home' position. Often you will not require the EOL function in graphics modes and so you will do all cursor movements by direct manipulation of 'ROWCRS' and 'COLCRS' before putting the next piece of data.

Whether you use EOL or not you must be aware that if the cursor moves beyond the bounds of the screen you will get an appropriate error code. When the cursor is on the bottom-right pixel of the screen the next put operation will cause the cursor to move to the start of the next line down, but this is not on the screen! In all modes except zero, another put operation would cause an error, so you would have to re-position the cursor first. In mode 0, when the cursor moves past the bottom of the last line, all lines are scrolled upwards, the last line cleared, and the cursor positioned at the start of it. This, however, makes it impossible to write to the last character on the screen without it scrolling!

Mode zero is treated differently to the other screens because it forms the basis of the 'E:' device, as we will see later. Firstly, all cursor movements are

CRACKING THE CODE continued

restricted to be between a left and a right margin value: 'LMARGIN' (52 hex) and 'RMARGIN' (53 hex). These are usually set to 2 and 39 respectively to give a two character margin on the left, but you can change these values at any time. Furthermore, you can temporarily over-ride the margin settings by manually changing the cursor position.

The major difference of mode 0 is that adjacent physical lines of text on the screen can become associated together to form logical lines of text. Logical lines are formed by writing text which extends beyond the right hand margin, causing it to start a new physical line. A logical line is terminated either by an EOL character or by the fact that it has exceeded three physical lines of text.

Whenever the screen is scrolled in this mode the top logical line is deleted which means that anything from one to three blank lines are formed at the bottom of the screen. The cursor is always positioned immediately after the last remaining logical line, i.e. anywhere from the first to the third line up from the bottom of the screen.

If the 'S:' device has been opened to allow reading, then characters or pixels may be read back from the screen at the current cursor position, for character modes the data is converted back into ATASCII. Unlike the put operation which will work in CIO's 'record' mode, the get operation should only be used in the 'character' mode, otherwise CIO will keep reading until the cursor goes out of range due to it not finding an EOL character!

The other standard CIO operations. CLOSE and STATUS, have no effect within 'S:'. As the screen is not cleared by a 'close', it is possible to re-open the device with the option specified for no-clear and thus have the screen preserved. There are two additional operations provided by 'S:' which are the DRAW (11 hex) and FILL (12 hex) commands, both of which work in any mode. Note that the codes for these commands in part fourteen were given transposed!

The draw command uses an algorithm which attempts to draw the best 'straight' line between any two points on the screen. The first point is specified by 'OLDROW' at 5A hex and 'OLDCOL' in low/high format starting at 5B hex. The last point is specified by the current cursor position in 'ROWCRS' and 'COLCRS'. You can either manually set both of these co-ordinates, or rely on the fact that after the current cursor position is used by 'S:' its position is automatically transferred into 'OLDROW' and 'OLDCOL'. This is particularly useful if you wish to draw a series of connected lines as you need only re-specify the current cursor's position each time. The variable 'ATACHR' at 2FB hex determines the data value which will be plotted by the draw command; as it always contains the value of the last character/pixel read or written it is not necessary to change it unless you want a different character or pixel colour.

CRACKING THE CODE continued

The fill command is really just a simple extension of the draw command. The cursor positions are set up, as described above, and the fill command is invoked. This causes a line to be drawn between two points, but at each pixel another line is drawn to the right for as far as the data remains zero. If there is no bounding pixel before the right hand edge of the screen then the process continues on the same line but from the left edge of the screen. In the worst case it will the entire line, but assuming you have drawn some pixels to the right of the line being drawn you will get the area filled in. The character or pixel colour used for the fill of the algorithm is obtained from 'FILDAT' at 2FD hex which you must set to the desired value.

The screen handler can produce the following erro codes:

8D hex: Cursor out of range for mode.
91 hex: Screen mode invalid.
93 hex: Not enough memory available for specified mode.

The last two of these can only be generated in response to an open command.

THE EDITOR HANDLER E:

The editor is really just an extension of the screen handler which only works in mode zero. Internally, the editor shares most of the screen handler's code and incorporates the keyboard handler as well. Due to this relationship, we only need to describe what is different about the editor compared to the screen handler.

As we mentioned above, the editor needs to be opened to use the screen handler in its split screen mode. When this is done, the screen handler manages the main area of the screen and the editor manages the four mode 0 lines at the bottom. Due to a conflict of the shared variables, the editor uses its own private variables for the split screen modes, thus allowing independent operation of both parts of the screen. The screen locations are: 'TXTROW' at 290 hex, 'TXTCOL' at 291 and 292, 'TINDEX' at 293 hex and 'TXTMSC' at 294 and 295 hex; these correspond to 'ROWCRS', 'COLCRS', 'DINDEX' and 'SAVMSC' respectively.

Opening the editor requires the specification of the device name, 'E:', and the first auxiliary byte for reading and/or writing (usually both). The second auxiliary byte is ignored as mode zero is always assumed.

Output to the editor behaves in exactly the same way as the screen handler except that certain characters have special meanings as follows:

- 1B hex: Escape; Sets the 'escape' flag. The next character will clear this and will be processed specially.

- 1C hex: Cursor up; Cursor up one physical line; wraps to bottom.

- 1D hex: Cursor down; Cursor down one physical line; wraps to top.

- 1E hex: Cursor left; Cursor left;

CRACKING THE CODE continued

wraps to right.

- 1F hex: Cursor right; Corsor right; wraps to left.

- 7D hex: Clear screen; Screen is cleared.

- 7E hex: Backspace; Character left of cursor is erased; does not go past the start of the logical line.

- 7F hex: Tab; Moves the cursor to the next tab point in the logical line or moves to the next line if one is not found.

- 9B hex: EOL; Moves the cursor to the start of the next logical line; scrolls if on the last logical line.

- 9C hex: Delete line; Deletes the entire logical line which contains the cursor; the lines below it are scrolled up.

- 9D hex: Insert line; Inserts a physical line on the containing the cursor; this can split the current logical line.

- 9E hex: Clear tab; Removes a tab point at the current column position in the logical line.

- 9F hex: Set tab; Sets a tab point at the current column position in the logical line.

- FD hex: Bell; Makes a 'bell' sound through the consol speaker.

- FE hex: Delete character; Deletes the character under the cursor and scrolls the rest of the logical line.

- FF hex: Insert character; Inserts a character to the left of the character under the cursor and scrolls the rest of the logical line.

The above characters, with the exception of EOL, can be printed if they are 'escaped'. This is done by sending the escape character and then the actual character you want printed. The escape character sets the 'escape' flag to indicate to the editor that the next character is to be printed, not acted upon. After the next character is printed, the escape flag is cleared. Following an escape character with another causes the escape character to be printed. If you are printing many of these escape characters then you can set the 'DSPFLG', at 2FE hex, to a non-zero value which causes every character to be treated as if it were escaped.

The tab points, mentioned above, are held in fifteen bytes starting at 'TABMAP', 2A3 hex. Each of the 120 bits in the tab map represents a column number in a logical line; if any bit is set then the corresponding column is a tab point and the cursor will move to it when the tab key is pressed. The tab points can either be set or cleared directly or by setting the cursor's position and sending the editor the appropriate character. Note that the tab map applies to every logical line on the screen; setting a point on one line will affect all tab points on all lines.

The CLOSE and STATUS commands have no effect within the editor; as with the screen handler, the display is not cleared when the close command is used.

CRACKING THE CODE continued

Where the editor differs substantially is in its handling od GET commands. A get character or record command will cause the editor to go into its 'editing loop'. This reads characters from the keyboard handler performing editing functions if necessary, such as deleting and inserting, until an EOL character is read. At this point all characters on the entire logical line containing the cursor, not necessarily the one which it was on when it was first called, are returned; one by one if it is a character get or all of them together if it is a record. Finally, a terminating EOL will be sent back after the whole line has been read, with the exception that trailing spaces are ignored.

There is one exception to the above rule and that is if text was previously printed without a terminating EOL, then only the characters from the cursor's original position onwards are returned by the get operation unless characters were typed to the left of that point. This is useful if you want to print a prompt before an input line as the prompt will not be returned. The exception to the exception (!) is that if the cursor is moved out of the logical line and then back in again, the whole line is returned as normal.

To allow you to read data from the screen directly rather than via the keyboard it is possible to use a 'forced' ead mode. If bit 0 was set in the first auxiliary byte when the editor was opened then get operations read entire logical lines directly from the screen. It is possible to directly alter the IOCB's auxiliary byte once the editor is opened, as it always re-checks this byte for each get operation. For instance, if you POKE 842,13 (IOCB 0) in BASIC, you will cause BASIC to continually read blank lines from the screen - unless, of course, you have put some command in the way, like RUN.

This 'trick' has often been used within BASIC programs to allow them to enter new program lines, such as data. The program will print text lines on the screen in the usual BASIC format and follow them with a CONTINUE or GOTO statement. It will then position the cursor above the new lines, switch on forced-read and execute a STOP statement. The new lines will be read as if they had just been typed and then the CONTINUE statement will start the program running again!

NEXT TIME

In the next issue we will publish the last part of CRACKING THE CODE. OHHHhhh! what a shame! It has been a great series and well written too. Does anyone know if there are more than 17 parts? if so let TWAUG know, so we can publish the rest of this article.

TABLE 1

Key-code to ATASCII conversion

Key Code	Key	Un-Shifted	Shift	Control
00	L	6C	4C	0C
01	J	6A	4A	0A
02	;	3B	3A	7B
03	Not Used			
04	Not Used			
05	K	6B	4B	0B
06	+	2B	5C	1E
07	*	2A	5E	1F
08	O	6F	4F	0F
09	Not Used			

CRACKING THE CODE continued

0A	P	70	50	10
0B	U	75	55	15
0C	Return	9B	9B	9B
0D	I	69	49	09
0E	–	2D	5F	1C
0F	=	3D	7C	1D
10	V	76	56	16
11	Not Used			
12	C	63	43	03
13	Not Used			
14	Not Used			
15	B	62	42	02
16	X	78	58	18
17	Z	7A	5A	1A
18	4	34	24	**
19	Not Used			
1A	3	33	23	9B
1B	6	36	26	**
1C	Escape	1B	1B	1B
1D	5	35	25	**
1E	2	32	22	FD
1F	1	31	21	**
20	,	2C	5B	00
21	Space	20	20	20
22	.	2E	5D	60
23	N	6E	4E	0E
24	Not Used			
25	M	6D	4D	0D
26	/	2F	3F	**
27	Inverse	**	**	**
28	R	72	52	12
29	Not Used			
2A	E	65	45	05
2B	Y	79	59	19
2C	Tab	7F	9F	9E
2D	T	74	54	14
2E	W	77	57	17
2F	Q	71	51	11
30	9	39	28	**
31	Not Used			
32	0	30	29	**
33	7	37	27	**
34	Backspace	7E	9C	FE
35	8	38	40	**
36	<	3C	7D	7D
37	>	3E	9D	FF
38	F	66	46	06
39	H	68	48	08
3A	D	64	44	04
3B	Not Used			
3C	Caps-lower	**	**	**
3D	G	67	47	07
3E	S	73	53	13
3F	A	61	41	01

Notes: 1. All values are in hex.
2. The symbol ** represents 'no value returned'.

TABLE 2

Screen Modes

Mode	Antic Mode	Pixels/Characters Across	Lines Not Split	Lines Split
0†	2	40	24	–
1†	6	20	24	20
2†	7	20	12	10
3	8	40	24	20
4	9	80	48	40
5	10	80	48	40
6	11	160	96	80
7	13	160	96	80
8	15	320	192	160
9	15/1	80	192	–
10	15/2	80	192	–
11	15/3	80	192	–
12†*	4	40	24	20
13†*	5	40	12	10
14*	12	160	192	160
15*	14	160	192	160

Notes: † = Text mode. 'Lines' refers to character cells.
* = Mode only available on XL/XE.
/1 = GTIA PRIOR bit 7 = 0, bit 6 = 1.
/2 = GTIA PRIOR bit 7 = 1, bit 6 = 0.
/3 = GTIA PRIOR bit 7 = 1, bit 6 = 1.

ALTERNATE REALITY: THE CITY

1. Getting Started, Mapping
2. Inns, Taverns, Banks, Shops,

THE CITY - Part 1

ALTERNATE REALITY begins with the first of six scenarios in THE CITY. Your character, after escaping from the spaceship in which he was abducted from his home, will find himself facing the Floating Gate in the City Square. Your character will have 3 packets of food, 3 flasks of water and some money. Although food and water will always be of concern in this game, you need not worry about them at this time. You do need a weapon.

To the north and south of this position are shops. To the west is a smithy; and to the east, one of the three banks. Near the bank there is a tavern, and near the smithy there is an inn. Turn to the west and enter the smithy. Make an offer on a dagger of slightly under whatever your character was given at the start of the game. If the smithy will not accept this offer, try making an offer on a stiletto.

Once armed, you are prepared as a beginning character to take on THE CITY. THE CITY is primarily a game of mapping and building your character for the scenarios that are supposed to follow.

Although the city square is reputed to be one of the safest parts of town, it is not the cheapest; and you will need to find cheaper Inns and Taverns in order to survive the initial developement of your character.

There are in THE CITY: 7 inns, 14 taverns, 3 banks, 15 shops, 4 smithies, 2 healers, and 12 guilds.

Inns provide a safe place for your character to rest and regain hit points. Inns will also allow you to check the time.

Taverns supply food and water. Food packets and water flasks are never offered for sale at the same time.

Banks will allow you to deposit your money in three types of accounts with increasing risk. They will also purchase gems and jewels.

Shops sell a wide variety of clothing which is of little importance in this first scenario. Shops also sell compasses for 5 silver pieces.

Smithies sell armor and weapons. As with the shops, prices vary and few items are cheap.

There are two healers. Healers can serve a variety of purposes, but are most useful for curing disease.

There are 12 guilds in THE CITY. In this first scenario your character may not join the guilds, but they still have their usefulness. Your first visit to a guild will improve one of your status or your chances of surviving. Guilds can also remove curses from weapons and armor. Prices for these services vary from guild to guild.

The map for THE CITY is 64

ALTERNATE REALITY: continued

squares by 64 squares. Square 1,1 is in the southwestern corner. When I give locations, I will be giving first the number of squares north on the map, the second number will be the number of squares east on the map. When I give the location of a place, the coordinates listed will be those of the actual place. If you must enter from a specific direction, and it is not an obvious door, these directions will be listed after the coordinates.

ALTERNATE REALITY: THE CITY - Part 2

The following is a list of Inns, Taverns, Banks, Shops, Smithies, Healers and Guilds, and their corresponding map coordinates.

INNS:
26,32 prices high*
25,33 prices high*
(* these are both the same inn)
24,33 reasonable
20,10 reasonable
4,32 very expensive
7,61 cheap
53,34 reasonable
55,29 cheap

More hit restoring benefits seemed to be gained by renting a room with a bath, but this is not a universal rule. Let your purse and needs decide.

TAVERNS:
30,40 expensive
20,33 reasonable, hours limited
25,8 reasonable, hours limited, enter from south
13,14 reasonable, special song at midnight
10,45 reasonable
3,61 cheap
31,61 reasonable,enter from east
32,59 to 32,60 south to 31,60
34,58 dues to join, expensive, enter from north
36,6 reasonable
36,7 reasonable
55,2 dues to join, limited hours
63,21 cheapest, enter by going north at 63,2, then go east to 64,21, then south.
54,34 dues to join, limited hours
57,53 reasonable, can be entered from south or west

Food packets and/or Pemmican will never be offered at the same time as water flasks. Menus change hourly. It is easiest to find food on even hours, and water on odd. This is not universally true, but a good guide. Almost all taverns sell food at midnight.

BANKS:
28,39 interest rates are lower but investment is safe
7,31 higher interest rates, more likely to lose money
62,3 higher interest, most risky, enter from south at 61,2
Banks that offer higher interest for your deposits are more likely to have bank failures.

SHOPS:
25,36
31,36
14,1 enter going west from square 15,6
13,4 enter going west from square 15,6
6,20
16,26
9,52
10,53
19,56
37,47

ALTERNATE REALITY: continued

56,34
57,38 enter from north
62,61
60,27
44,21-22
38,10

Shops carry a wide variety of clothing which has no importance in THE CITY, but will become important in THE DUNGEON. Shops have different hours. All shops sell compasses for 5 silver.

SMITHIES:
28,33
10,55
35,51
33,20 enter from north
Smithies have different hours. Goods change, prices change, and almost everything is expensive.

HEALERS:
20,5
30,30
Healers tend to be open for business on ODD hours. Repeated visits in one day will cause prices to increase.

GUILDS:
44,35 Thieves Guild, increase skill, enter from west. 3,56 Assassins Guild, increase stealth, north from 2,57 to south from 4,56. 15,48 Red Wizards Guild, increase strength, north from 13,47 east to 14,48, north 22,34. Dark Wizards Guild, increase charm 48,19. Blue Wizards Guild, increase speed, enter from west 12,28. Star Wizards Guild, increase hit points and strength 15,6. Physicians Guild, increase hit points, enter from west 5,3. Light Wizards Guild, increase wisdom, enter from west 60,51 Wizards of Chaos Guild, increase charm, enter from east 50,58. Guild of the Order, increase intelligence 50,62. Wizards of Law Guild, increase wisdom 43,12. Green Wizards Academy, increase stamina, enter from north. Guilds will raise your status on your first visit to them. Guilds can remove curses from armor and weapons. Costs will vary from guild to guild. Prices can be as low as 2,000 coppers or as high as 11,000 coppers.

ALTERNATE REALITY: THE CITY - Part 3

POTIONS:
COLOR TASTE SIP EFFECT
AMB PLN C Cure Poison
AMB PLN DD Poison
AMB SOUR S Spirits
AMB SOUR S Beer
BLK ACID C Invulnerability Fire
BLK ALK C Invulnerability Water
BLK BIT C Invulnerability Mental
BLK BIT U Delusion
BLK DRY C Invulnerability Power
BLK PLN C Invulnerability Sharp
BLK PLN C Invulnerability Blunt
BLK PLN C Fleetness
BLK SALT C Invulnerability Air
BLK SOUR S Beer
BLK SOUR DD Strong Poison
BLK SOUR C Invulnerability Earth
BLK SWT C Invulnerability Cleric
CLR ACID S Cure
CLR ACID C Water
CLR ACID DD Acid
CLR ACID C Cleanse
CLR BIT C Unnoticeability
CLR DRY C Mineral Water

ALTERNATE REALITY continued

CLR DRY C Invisibility
CLR PLN C Water
CLR PLN C Invisibility
CLR SALT S SaltWater
GRN SOUR C Heal Minor Wounds
GRN SWT DD Ugliness -1 Charisma pt.
ORG BIT S Inebriation
ORG SOUR C Protection+2
ORG SWT C Protection+1
ORG SWT DD Dumbness -1 Int. pt.
RED ACID S Vinegar
RED BIT C Strength
RED DRY S Wine
RED SWT C Treasure Finding
RED SWT DD Deadly Poison
RED SWT C Fruit Juice
SLV BIT D Weak Poison
SLV BIT C Intelligence
SLV PLN C Cure Major Wounds
SLV SWT C Charisma
WHT ALK C Milk
WHT ALK C Healing
WHT ALK DD Poison
WHT BIT DD Slowness
WHT SALT C Heal All
YLW BIT C Noticeability
YLW DRY DD Weakness -1 Str. pt.
YLW PLN C Cure Wounds

LEGEND
Colors: RED=Red, BLK=Black, SLV=Silver, WHT=White, YLW=Yellow, CLR=Clear, ORG=Orange, AMB=Amber, GRN=Green.

Tastes: SWT=Sweet, SALT=Salty, SOUR=Sour, BIT=Bitter, ALK=Alkaline, ACID=Acidic, PLN=Plain, DRY=Dry.

Sips: D=Danger, C=Caution, S=Safe, DD=Dangerous, U=Unsure.

Potions can be helpful or harmful. The higher your wisdom and intelligence, the more easily you will be able to identify the potions. Some potions such as Fleetness and Protection+1 and Protection+2 have a greater frequency of occurrence than do the others. What a potion will be is randomly decided by the program when you unseal the potion. You can increase your control over potions by saving them, saving your character, making backup copies,re- entering the game and trying them. If you are not satisfied with the potions you got, you can restore and try again. Treasure Finding will greatly increase your ability to find money, potions and weapons!

ALTERNATE REALITY: THE CITY is published by Datasoft. The list of Potions was submitted by Don Wyman.

ALTERNATE REALITY: THE CITY - Part 4

GAME HINTS:

1) Save your character often to a separate backup disk!

2) Have (or get) a fast copy utility. Do NOT use the backup utility that comes with the game.

3) Copy your character to several disks when: A) you have many potions or B) you are about to increase a level. Enter the game with each of these disks, keeping notes on what potions you found or what stats were increased. Use the best of the ones you tried, or recopy and try again!

4) Avoid Brown Molds, Black

ALTERNATE REALITY
continued

Slimes, and Giant Rats as often as possible until you are either rich and can afford trips to the healers, or can defeat these monsters. Tricking them is a successful means of defeating these disease giving creatures, IF you have high enough intelligence.

5) Do not carry more food, water, gold and gems then you really need to. Excess weight will cause your character to become weary sooner and reduce your adventuring time between stops at the inns.

6) Do not bother to examine or taste potions, just sip. Avoid too many protection potions as this will cause your character to crash.

7) While still a lower level character, avoid being out at night or in the rain.

8) When finding a weapon, always equip it as a secondary weapon in case it is cursed.

TWAUG ANNOUNCEMENT

THE TWAUG PD LIBRARY

Since TWAUG first released issue 1 newsletter and disk in January 1993, we have built up a PD library containing very nearly 500 titles. We are always on the search for new titles to add to the library, and hope to cover as many aspects of the Atari 8-bit as possible.

TWAUG ANNOUNCEMENT
continued

Although, the money we receive for subscription to TWAUG goes a long way towards the running cost, we do depend a lot on the sales of our PD disks to keep things running smoothly. We have always intended to keep the cost of subscribing to TWAUG as low as we possibly can, and we have no plans to increase the subscription in the near future unless it becomes absolutely necessary.

As you will have noticed, we are now increasing the size of the text in the newsletter which in turn means more pages. In time, we hope to be able to add a wider range of topics which will mean even more pages.

Although our list of subscribers grows with each new issue, we do not receive as many orders for PD software as we would like to see. Our thanks goes to all of those who have purchased PD disks from the library, your support is greatly appreciated. However, we would like to see more of you if possible, buying the odd one or two disks from the library and help us to work on more ideas that we have in providing more for our subscribers. We would like to be able to buy the right to some commercial software. At the moment, I am trying to track down some programs that have not been seen for some time now and I am sure many 8-bit users would be very interested in, but I would like to have money in the bank to be able to buy the copyrights.

TWAUG ANNOUNCEMENT continued

Have a good look through all your back issues of the PD library and see if there is something that you feel you may be able to make use of. If you no longer have the updates, then just drop us a line and we'll send you a printed copy of the full PD library list.

THE TWAUG PRINT SHOP BOOKLET.

We believe that we have one of the largest collection of Print Shop icons, fonts and borders for the 8-bit. Due to recent additions to this catalogue, John is busy reprinting the booklet to bring it up to date. There will be quite a few extra pages in the new booklet, therefore, we will have to, unfortunately, increase the price. The Print Shop booklet will now cost £1.25, but it is worth every penny as it can not only help you decide which disks to buy, but it is also a very good quick reference guide when you are looking for one particular icon, font or border on the disks you have already purchased from the library.

THE ATARI 8-BIT BOOK LIBRARY.

In issue 15 of our newsletter, we announced that we were starting an Atari 8-bit book library. I am glad to say that it is now up and running, and we have a printed catalogue ready to send out to anyone who may be interested in having a look to see what books we have.

Many of the books in the library are very rare and cover a wide range of topics including programming in BASIC, Pascal, 'C', Logo and also Machine code. There are books to suit both beginners and advanced programmers. There are also books for those who just like to type in programs including some for children.

At the moment, we have over 60 books in the library, but more books are on the way from the U.S.A. Why not send for our catalogue, it costs nothing to have a look.

To become a member of the book library will cost £5.00, this is to give us money to buy even more books as and when they become available to us. Books can be hired for a period of one month for the cost of postage and packing which will depend on the weight of the book. It will range from £1.00 to £5.00. Some of the books we have are A4 size and contain over 400 pages. The hire charge for each book is given at the end of each book description in the library catalogue.

THE OL'HACKERS ATARI USER GROUP INC.

O.H.A.U.G. is an all 8-bit user group in the State of New York U.S.A., they are producing a bi-monthly first class informative newsletter on disk.
The disk is double sided full of news, views, articles and bonus games and/or utilities. The disk has its own printing utility which lets you read the content on screen or make hard copies. A large PD Library is also available.
For more information contact:
Mr.A.Pignato
O.H.A.U.G.,3376 Ocean Harbor Drive, Oceanside, N.Y.11572, U.S.A.

LETTER SECTION

Dear TWAUG,

Before I start I must apologise for the name Commodore Amiga and C64 being mentioned in this letter. No, I haven't posted this to the wrong mag but it gave me an idea for our beloved Atari classic's.

Many of you will have read in the news about the sale of Commodore to Escom, who recently bought the Rumbelows electrical chain and their pledge to re-introduce the A1200 and A4000.
Escom's top man, Bernard van Tiemen, is said to be planning to dig up the old C64 computer and sell it in Eastern Europe!

My immediate first thought was..."What has Eastern Europe done to deserve that being dumped on them?" My thoughts then drifted to our Atari 8-bits.

If Escom can see a marketplace for a computer so inferior to our Atari in Eastern Europe then why doesn't Atari leap in there first and give them something worthwhile.

If the 8-bit is as dead as Atari think then why is a firm like Escom prepared to take such a huge risk? Just imagine how our Atari Classic's would get a new lease of life, if only Atari would put their thinking caps on and decide to re-release the 65XE and 130XE.

by Mr.T.L.Bingham.

Dear David, John and Max,

I just thought I'd drop you a line to say thanks for printing my articles "Quieten the 1050" and "XIO Explained" in issue 14 & 15 of the TWAUG newsletter. It really was quite a buzz seeing my name in print and I urge other subscribers who are thinking of writing an article to do so because, to coin a phrase, 'If I can do it, so can anyone'. Actually at the moment I'm in the process of writing some further material for you which I shall be submitting for publication over the next few weeks or so.

Whilst I'm writing can I just take this opportunity to introduce you to Chaos! Computers, a new company which I've set-up to provide additional support to Atari Classic owners. Chaos! Computers won't be competing with any existing 8-bit outlet, as we will be concentrating more on making available new or relicensed hardware and software at the lowest possible prices. The first product we have announced is 'The Hyper Drive' which is similar in many ways to the old Happy/Lazer enhancements. The introductory price is £30.00 which includes the Hyper Drive P.C.B., Version II Software Diskette and the full 28-page A5 Installation and Operating Manual. We aim to provide back-up for all our products, as well, and we hope to be able to release disks of 'PDB' or Pre-Determined Back-Up files, which of course will be of use to not only Hyper Drive owners, but also to Happy and Lazer users too. We've also

LETTER SECTION
continued

managed recently to get hold of some older software titles, due to abulk purchase, which haven't been available for quite some time, details of these programs will be announced shortly-so watch this space, as they say.

I'd be grateful, if you print this letter, if you could include the above address so that if anyone wants details of any further products from Chaos! Computers, or indeed wants to order anything that I've mentioned above, they can get in touch. Thanks!

Once again can I say thanks for providing a much needed forum for all us die-hard Classic users. Wishing you all the best.

Yours Atarily,
Paul Hollins
Chaos! Computers
PO Box 30
Manchester M19 2DX

CONTENT

This issue disk has a fantastic selection of programs for you to enjoy. I (Ed) particularly like Deskcalc and Maillist, both are Basic programs and you'll find both docs on the disk.
The DESKCALC.BAS has all the features a calculator has only this computer calculator has extra features. When you've done all your calculations you can add comments and then make a hard copy of all your sums with the added comments.

CONTENT cont.

It isn't difficult to operate this calculator, as a matter of fact it is quite simple, but making a hard copy of the documantation makes it doubly easy by just referring back to the doc. The amount of pages to print this document is three pages plus about four lines on the fourth page. The printer setup for that is 58 lines per page and 60 characters per line, the easiest way is to load this text into a word processor and then print it.

The other program I like is the MAILLIST.BAS. This is a database for your very important addresses with a difference. Fitting-in a lengthy address is a doddle, there are no fields to out manoeuver, just type-in your address and after each line type Return, after the last line type Return twice. When the program has loaded your are presented with seven options. You have also a choice to use a word processor to setup your addresses and transfer them to the database.

As always all programs run from the menu on side 1 of this issue disk. There are three more games on side 2 and they too run from the menu from side 1, a prompt tells you to flip the disk. Besides the three games on side 2 are other Turbo Basic files, these programs go with the article "The PRC Construction Set" on page 15 and 16. Take the Turbo Basic programs of the issue disk and put them onto a disk with Turbo Basic to run them.

FOR SALE

4 800XLs, no power supplies, £10 each plus P&P.

130XE with 1050 disk drive including all leads and power supplies £75 plus P&P.

Tape deck with about 30 tapes, a quantity of computer/disk drive spares, any reasonable offer.

Please contact:
Mr. Copeland
51 Hollings Street
Fenton
Stoke on Trent, ST4 3HS
TEL. 01782/333044.

FOR SALE.

800XL (rev C), 1010 tape deck includes all leads and power supplies plus a few cassettes £25.

130XE with XC12 cassette deck, two 1050 disk drives, Ketec printer interface and all power supplies and cables.

Approx 200 disks full of progs some games but mostly utilities/serious software. Approx 50 blank disks, 2 lockable boxes, lots of books and magazines £120 or will split.

I also have a Hercules 12 inch amber screen monitor (needs attention), with appropriate cable, would suit an ST. I also have the matching video card for the PC. Open to offers.

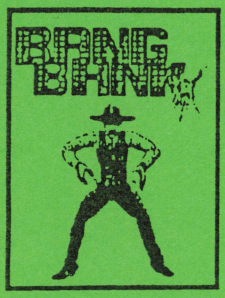
Stamped Addressed Envelope for list to:

Mr. John Bunting
101 Windsor Drive
Wingerworth
Chesterfield, S42 6TQ
TEL. 01246/238187.

ANNOUNCEMENT

Some of you will have noticed that the "DIGI DEMO" on disk 2 side B of issue 15 does not run on a standard 800XL or 65XE, unfortunately we never noticed it until someone got in touch with TWAUG. DIGI Demo which is boot load needs more then 32K of memory and therefore only runs on 130XE's or upgraded 800XL or 65XE's.

As it so happens we at TWAUG all use upgraded machines and didn't notice this problem. From now on we will keep a standard XL handy for checking future programs.

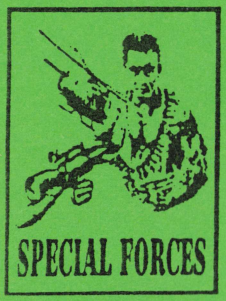


Are You Still Complaining about lack of New software for your Atari 8 bit ? If you are you don't know about Mail Order From

MICRO-DISCOUNT

265, CHESTER ROAD.
STREETLY.
WEST MIDLANDS.
B74 3EA.
ENGLAND.

TEL 021 353 5730
FAX 021 352 1669



ADAX *HANS KLOSS*

DARKNESS HOUR

TWAUG NEWSLETTER

The Atari Classic Programmer's Club

This is to inform you membership is now only available on yearly subscriptions. The life membership option is no longer available. A member has access to various programmer services, a helpline, regular printed newsletter and discounts off our software?

Current UK membership cost is:

12 months: £6.00

Overseas members most welcome. If you require more details then please send an SAE to the address below. Overseas Atarians, please send two International Reply Coupons (available from your post office) for more information.

Still Available:

Swift Spreadsheet (Standard) - £9.95

Swift Spreadsheet (New) - £12.95

(NOTE: New version includes a revised 40 page A4 manual. For details of the standard version and of the program itself, please refer to the review in issue 67 of New Atari User).

Available Soon:

Menu Print: Still under development and currently undergoing its THIRD rewrite! We apologise for the delay but we expect the programming to be definitely completed by the end of September. More details will be published when ready.

THE ATARI CLASSIC PROGRAMMER'S CLUB
Pen-Tyddyn
Capel Coch
LLangefni
Anglesey, Gwynedd LL77 7UR
Wales

CHAOS! COMPUTERS

PO BOX 30 MANCHESTER M19 2DX
Telephone: (0161) 737 1946

THE HYPER DRIVE

Upgrade your ATARI 1050 disk drive with a HYPER DRIVE enhancement from CHAOS! COMPUTERS.

The HYPER DRIVE is an easily installed hardware & software package for the ATARI 1050 which will enable your disk drive to back-up most disks protected by unreadable or badly formatted sectors. Most copied disks can then be loaded on any 1050, whether enhanced with a HYPER DRIVE or not.

The HYPER DRIVE enhancement also offers fast reading, writing, formatting and copying in single, medium or true double density formats (i,e. it is compatible with RANA, PERCOM and INDUS double density drives, and will read U.S. DOUBLER type format).

Fitting the HYPER DRIVE coudn't be simpler and requires no special tools or soldering. It simply plugs into socket on the 1050 circuit board. And with our VERSION II software package and full 28 page manual, it is one of the most versatile disk drive enhancements /copiers you can buy.

HYPER DRIVEs are available exclusively from CHAOS! COMPUTERS at a special introductory price of just £30.00 each.

Please make Cheques/Postal Orders payable to 'P. HOLLINS'.

Prices are subject to change, from time to time, due to component costs, so wherever possible please 'phone to check.